Module 5 B1

Digital Techniques
Electronic Instrument Systems

5.1 Electronic Instrument Systems

5.2 Numbering Systems

5.3 Data Conversion

5.4 Data Buses

5.5 Logic Circuits

5.6 Basic Computer Structure

5.10 Fibre Optics

5.11 Electronic Displays

5.12 Electrostatic Sensitive Devices

5.13 Software Management Control

5.14 Electromagnetic Environment

5.15 Typical Electronic / Digital Aircraft Systems

5. Study Questions

5.1 Electronic Instrument Systems

Introduction

Regarding the Electronic Instrument System of modern aircraft, it is difficult, if not impossible, to give a description without referring to specific Aircraft types such as Airbus A320, MD11 etc.

The Airbus A320 was the first civil aircraft with an all-electronic flight deck indication system and now many aircraft, both large and small - including helicopters, have *Cathode Ray Tube* flight deck displays. (or even *LCD* Displays for the newest aircraft types).

In this submodule follows a general introduction with a view of the *transition* from old fashioned cockpit instruments to the modern Electronic Instrument System. A short description of the Electronic Instrument System of the MD11 is given.

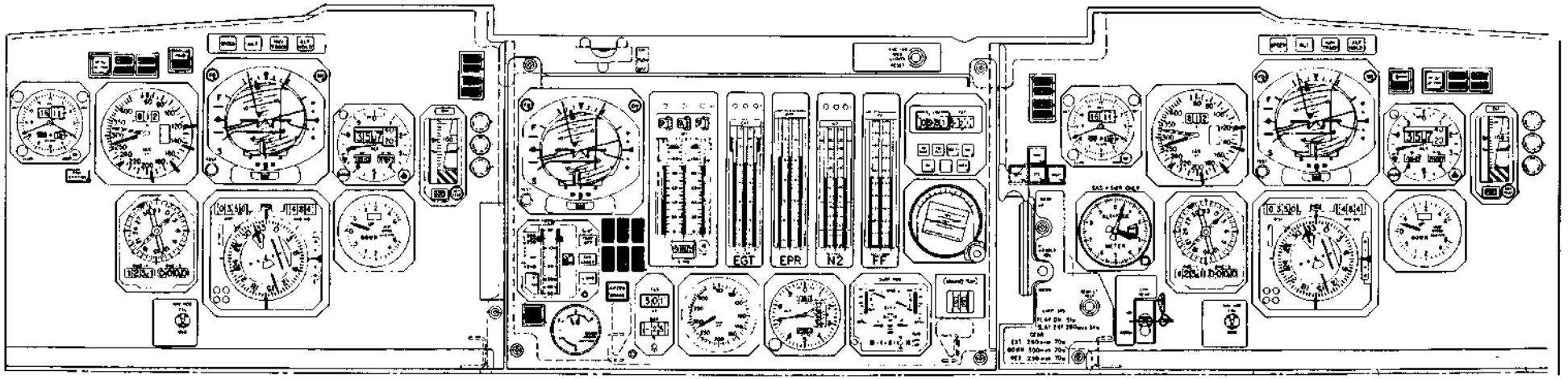
After that, to give a detailed description of the typical system arrangement of modern Electronic Instrument Systems, two Aircraft types are chosen as examples. These are the Airbus A320, that represents the larger modern civil aircraft, and the Embraer 145 that represents the smaller modern turbofan transport aircraft.

If you work on a different type of aircraft, the system may be different from the ones described in this submodule, and may also have different names for the components used. But after studying the two Aircraft Electronic Instrument system, that will be discussed on the following pages, you will have a good base and the necessary knowledge for the further study of aircraft type rating courses.

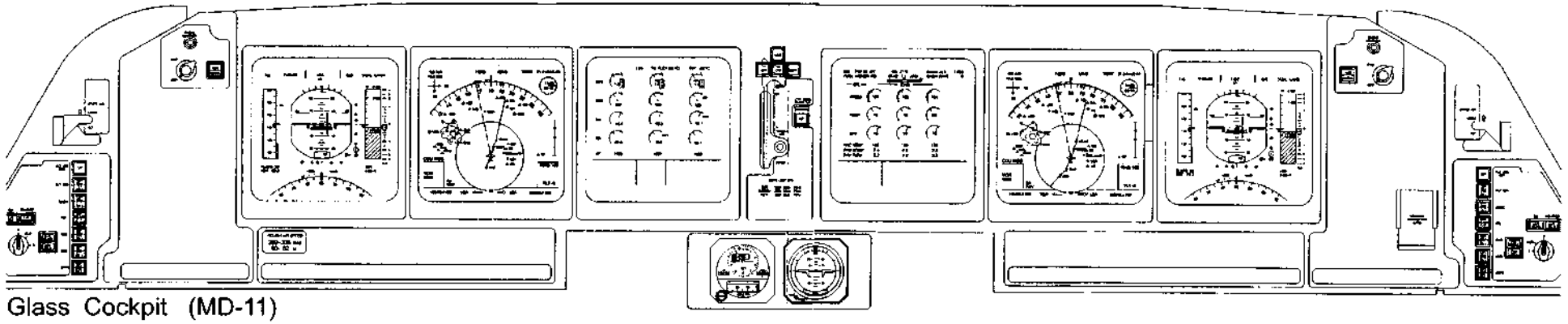
Digital electronics has opened an extreme *wide* door for new *developments*. Cathode Ray Tubes (CRTs) are used as multifunctional displays in the modern "glass cockpits". A single display replaces a number of mechanical analog type indicators.

In addition of displaying instrument indications, CRTs may be used to display check lists and operational history of the portions of a system which are showing trouble, *suggest* corrective action, and display any performance reduction caused by the malfunction.

Figure 1: Analog vs. Digital Cockpit Instrument Systems



Conventional Instrumentation (DC-10)



Glass Cockpit (MD-11)

Electronic Instrument System - General

The Electronic Instrument System (EIS) is an avionics system connected with most of the aircraft systems to *carry out* the two following main display functions:

Electronic Flight Instrument System & Electronic Centralized Aircraft Monitor.

EIS = EFIS + ECAM

The **Electronic Flight Instrument System (EFIS)** enables all flight and navigation informations to be communicated to the crew. The cathode ray tube display units group together the parameters from the following conventional instruments as installed on the DC-10. Listed in the right column are the *corresponding* EIS display units, featuring those indications on an MD-11, see block diagram to the right.

Attitude Director Indicator, ADI	PFD
Horizontal Situation Indicator, HSI	ND
Altimeter	PFD
Mach *Airspeed* Indicator	PFD
Weather Radar indicator	ND
Flight Mode Annunciator, FMA	PFD

The **Electronic Centralized Aircraft Monitor (ECAM)** enables the visualisation of the information concerning normal and abnormal *state* of the aircraft systems (EAD and SD).

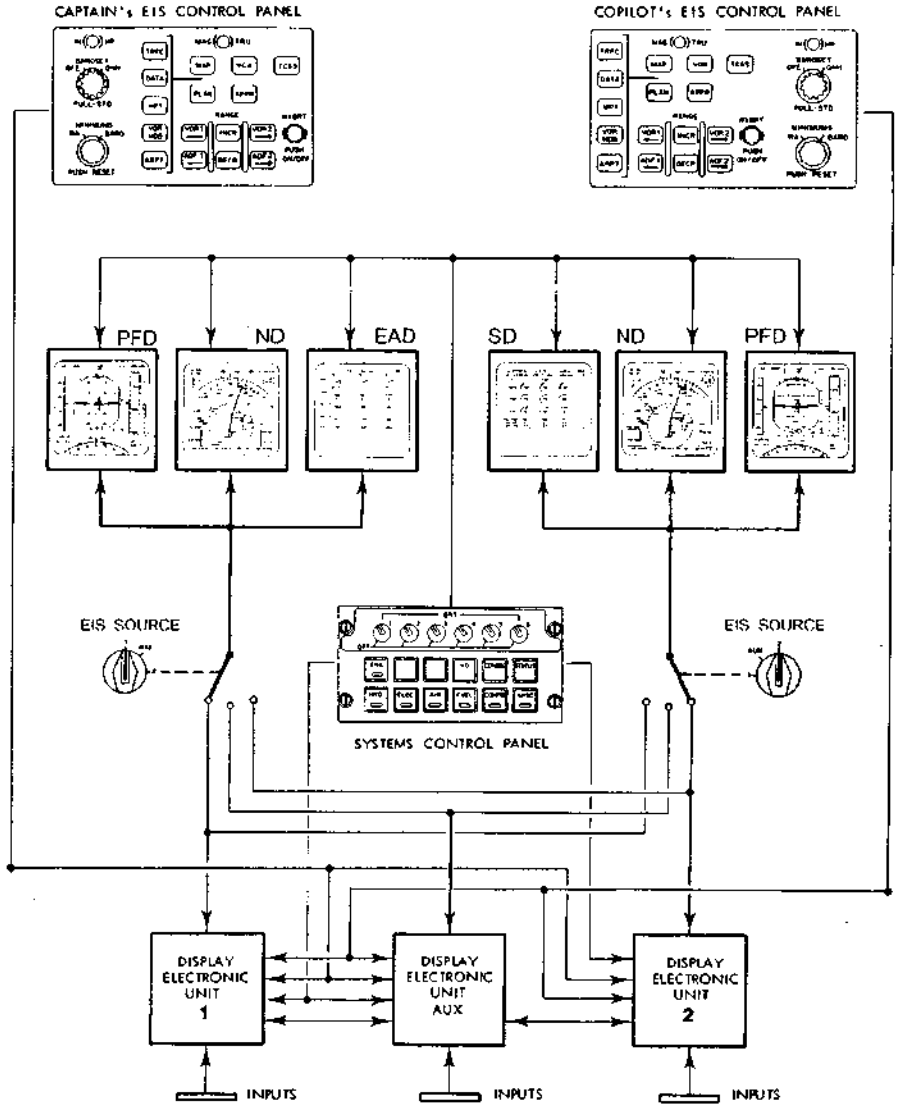
In other aircraft it's named **E**ngine **I**ndication And **C**rew ***A**lerting* **S**ystem (EICAS).

Note:

As a typical system arrangement the simplified block diagram on the right side shows the EIS interface of an MD-11.

A short description of the components is given in the introduction on the next page.

Figure 2: EIS Interface Block Diagram of an MD11



MD-11 Electronic Instrument System Arrangement

The EIS components are:

- six Display Units (DU), mounted on the instrument panel,
- two Source Input Select Panels (SISP), mounted on the instrument panel,
- two Electronic Flight Instrument Control Panels (ECP), located in the glareshield,
- one Systems Control Panel (SCP), located in the pedestal,
- one Remote Light Sensor (RLS), located on the glareshield, and
- three Display Electronic Units (DEU), located in the Electronics Bay.

  Note:
  Each DEU manages all EIS functions (EFIS on PFD and ND and ECAM on EAD and SD) and - in case of dual DEU faults - a single DEU could control all six display units *simultaneously* (see EIS SOURCE switching on the *previous* block diagram).

The EIS displays *consist* of:

- PRIMARY FLIGHT DISPLAY (PFD)

  Displays *attitude*, *airspeed*, *barometric altitude*, radio *altitude*, vertical speed, *heading*, vertical and *lateral deviation*, limits, configuration, and *flight modes*.
- NAVIGATION DISPLAY (ND)

  Displays a *pictorial* representation of the aircraft position and *relevant* waypoints, navaids, airports and other flight plan background data and *weather radar* or terrain displays (the last available, if an *Enhanced* Ground Proximity Warning System is installed).
- ENGINE/ALERT DISPLAY (EAD)

  Displays the primary engine data N1, N2, EGT, (EPR *optional*), fuel flow and *alert* messages.
- SYSTEMS DISPLAY (SD)

  Displays either secondary engine data (engine oil temperature, pressure, and quantity), or systems pages like hydraulics, electrical, air (environmental), fuel, configuration and miscellaneous (not categorized systems) with the *associated alerts* and consequences.

  The SD can also present the aircraft status and consequences, resulting from all *alerts*, and it may be used to display an ND image in case of a DU fault.

EIS Controls

The PFD and ND are controlled through an EFIS Control Panel (ECP), one for each pilot, located to the left and to the right of the Flight Control Panel (FCP). All of them are part of the Glareshield Control Panel (GCP).

The Systems Display is controlled through a Systems Control Panel (SCP) located in the pedestal aft of the throttles.

A320 Electronic Instrument System Presentation

General

The Electronic Instrument System (EIS) *comprises* captain's and copilots EFIS (Electronic Flight Instrument System) and the ECAM (Electronic Centralized Aircraft Monitor) system.

EIS data are presented on 6 identical color Cathode Ray Tubes (CRT) type Display Units (DU).

EFIS

Each pilot's EFIS includes 2 DU's, a Primary Flight Display (PFD) and a Navigation Display (ND).

ECAM

The ECAM data are displayed on an upper DU, called Engine and Warning Display (E/WD) and a lower DU called System Display (SD).

Display Management Computer

EFIS and ECAM DU's are driven by three identical Display Management Computers (DMC). Each DMC has independent EFIS and ECAM channels and is able to drive *simultaneously* one PFD, one ND and either ECAM E/WD or SD.

The EFIS channel *acquires* and *processes* signals received from navigation and auto flight systems and generate the images to be displayed on PFD and ND.

The ECAM channel *acquires* and *processes* signals received from sensors and computers via two SDAC's (System Data *Acquisition* concentrator) and from two FWC's (Flight Warning Computer) and generate the images to be displayed on E/WD and SD.

In normal operation, DMC-1 supplies data to captain's PFD and ND (EFIS DU's) and the E/WD (upper ECAM DU).

DMC-2 supplies data to copilot's EFIS DU's and the SD (lower ECAM DU).

DMC-3 is in stand-by.

If DMC-1 or 2 fails (indicated by a diagonal line on the *corresponding* DU's), an EIS DMC switching selector allows to replace the failed DMC by DMC-3.

In case of a DU failure, the remaining DU's can be *reconfigured* to ensure the display of all required information.

EIS data are presented on 6 identical Display Units (DU). The layout of the 6 DU's will be presented as follows:

2 DU's are installed side by side in front each pilot.

They display flight and navigation data on each pilot instrument panel.

In normal configuration:

The outer DU will be allocated to the Primary Flight Display (PFD) function and the inner DU will be allocated to the Navigation Display (ND) function.

- **EFIS DU's = PFD + ND**

2 DU's are installed on the center instrument panel, one above the other.

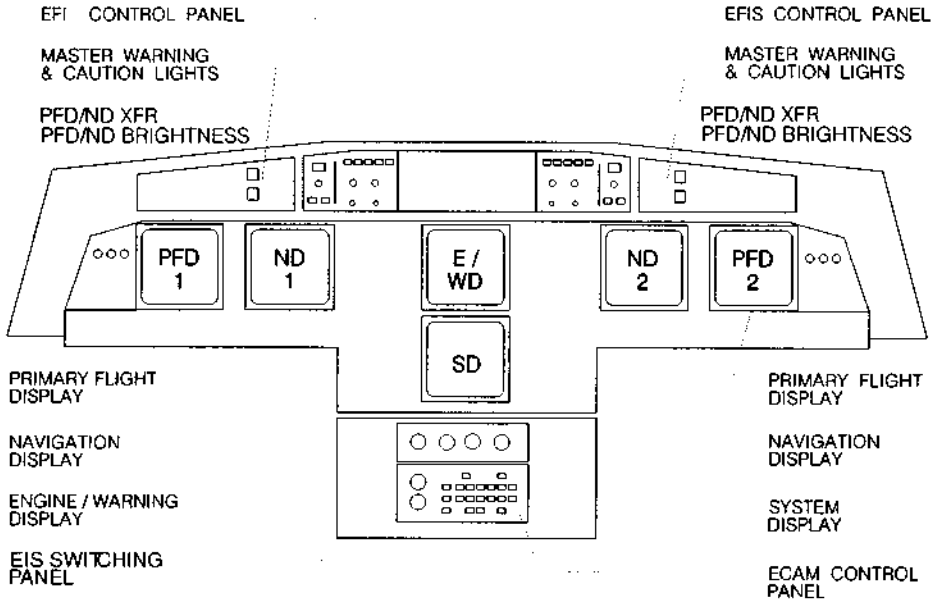
The upper DU will be allocated to the Engine/Warning Display (E/WD) and

the lower DU will be allocated to the System Display (SD).

- **ECAM DU's = E/WD + SD**

Figure 3 illustrates the cockpit arrangement of the EIS system

Figure 3: Cockpit Arrangement



DU Reconfiguration

PFDU failed or selected OFF

In case of a PFD Unit failure, the PFD image is automatically transferred to the ND Unit.

This transfer can also be *achieved* manually by pressing the PFD/ND XFR push button, which will cross-change the images between the PFDU and the NDU.

NDU failed or switched OFF

In case of a NDU failure, the ND image can be displayed on the PFDU *instead* of the PFD image by pressing the PFD/ND XFR push button.

Upper ECAM DU failed or switched OFF

The E/WD image is automatically transferred to the lower ECAM DU.

The SD images may be recovered either:

- On a NDU by use of the ECAM/ND XFR selector on the SWITCHING panel

or

- On the lower ECAM DU *instead* of the E/WD image by depressing and holding the desired system page key on the ECAM control panel. After 30 seconds, the E/WD will reappear.

Lower ECAM DU failed or switched OFF

The SD image may be displayed either:

- On NDU by means of the ECAM/ND XFR selector on the SWITCHING panel

or

- On the upper ECAM DU *instead* of the E/WD image by depressing and holding the desired system page key on ECAM control panel. After 30 seconds, the E/WD will reappear.

Both ECAM DU's failed

- The E/WD image may be displayed on the NDU by use of the ECAM/ND XFR selector on the SWITCHING panel.
- The SD images may be displayed *instead* of the E/WD image on a NDU (ECAM/ND XFR selector on CAP or F/O) by depressing and holding the desired system page key on ECAM control panel. After 30 seconds, the E/WD will reappear.

Figure 4: DU Reconfiguration

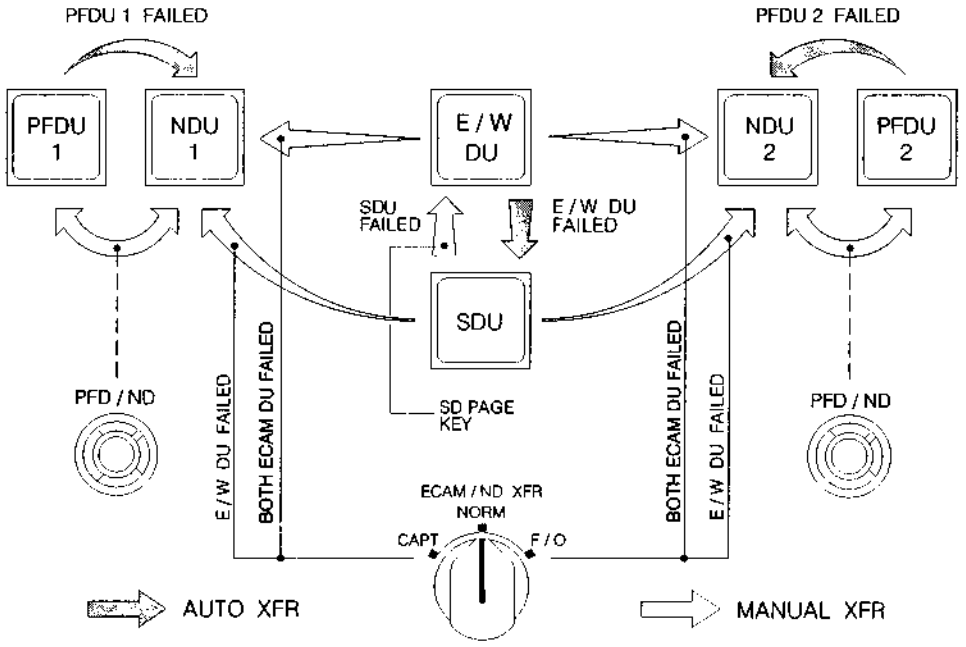
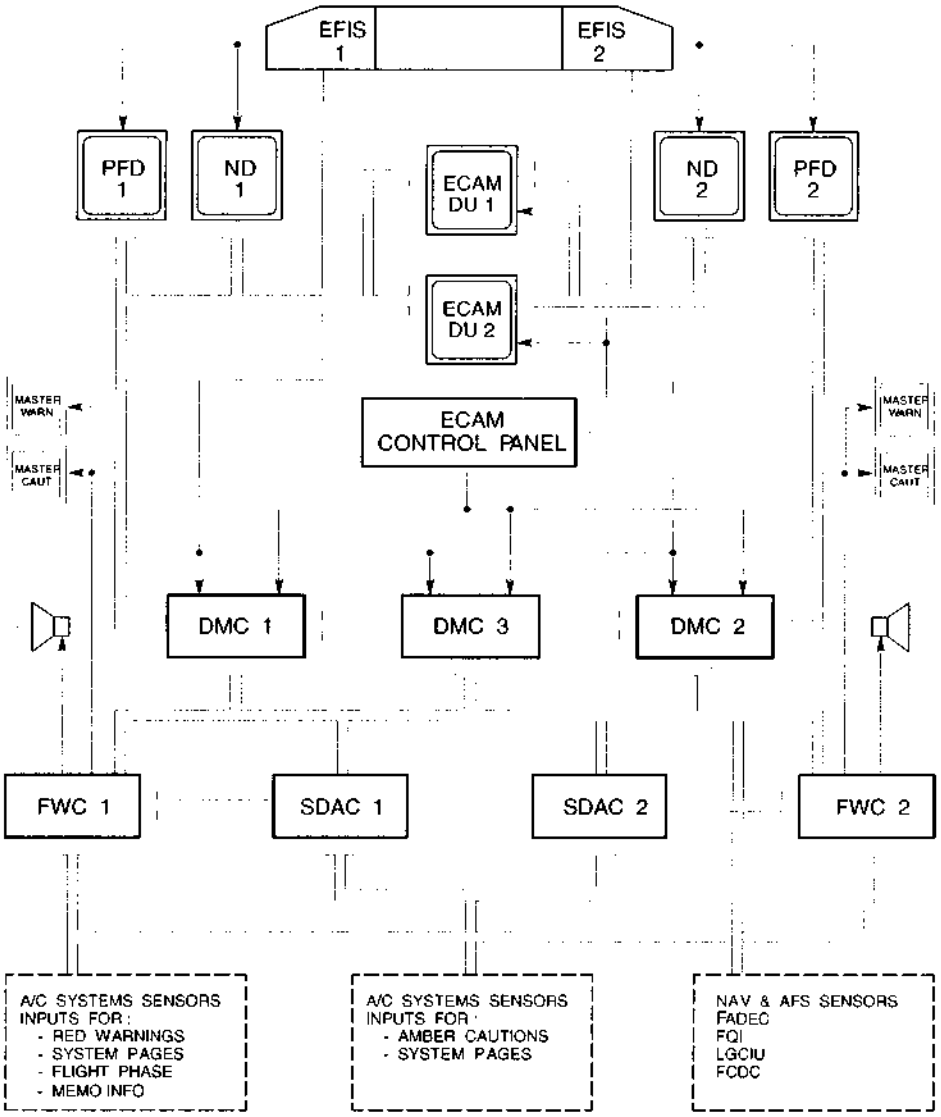


Figure 5: EIS A320 System Architecture



Main Components and Subsystems

EFIS

The Electronic Flight Instrument System (EFIS) presents to the flight crew the data necessary:

- To control the aircraft by means of two Primary Flight Displays (PFD).
- To navigate by means of two Navigation Displays (ND).

PFDU's and NDU's are driven by the EFIS channels of the DMC's.

Primary Flight Display (PFD)

The Primary Flight Display *provides* mainly:

- *Attitude* and *guidance* information.
- *Airspeed.*
- *Altitude* (*baro* and radio) and vertical speed
- *Heading* and *track.*
- FMGS modes (*flight mode* annunciation).
- Vertical and *lateral deviations.*
- Radio Nav information (ILS, DME).

Main parameters such as *attitude*, *heading* and *altitude* are monitored by the FWC's.

Normally, a grey background is displayed on speed, *heading* and *altitude* windows. In case of avionic ventilation failure, the grey background is suppressed in order to limit PFDU power consumption and to prevent them from overheating.

Figure 6: PFD General Arrangement

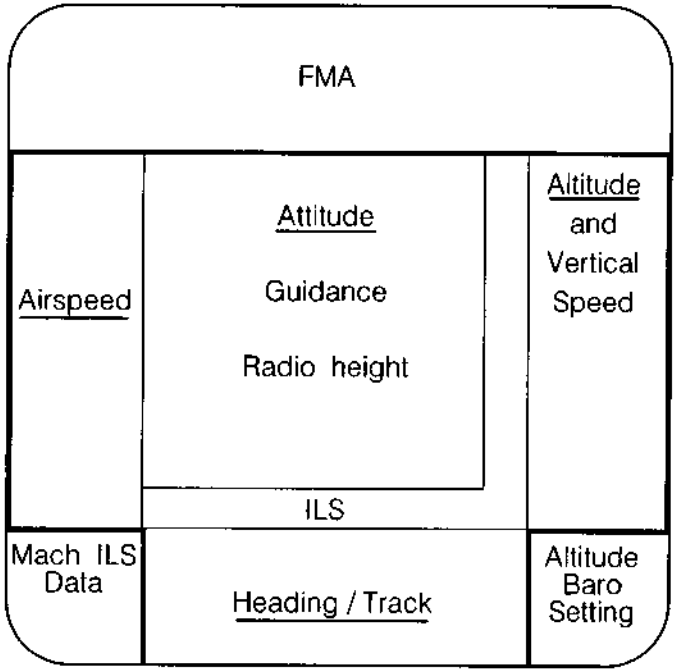
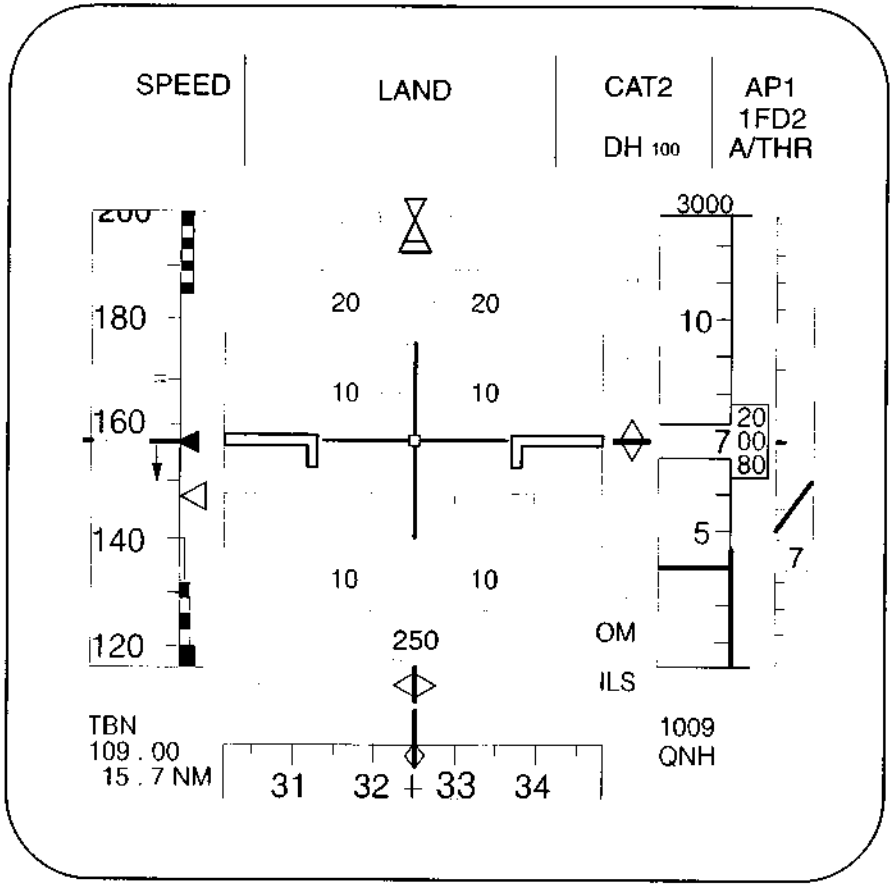


Figure 7: Example of the Primary Flight Display



Navigation Display (ND)

Five different display modes are available:

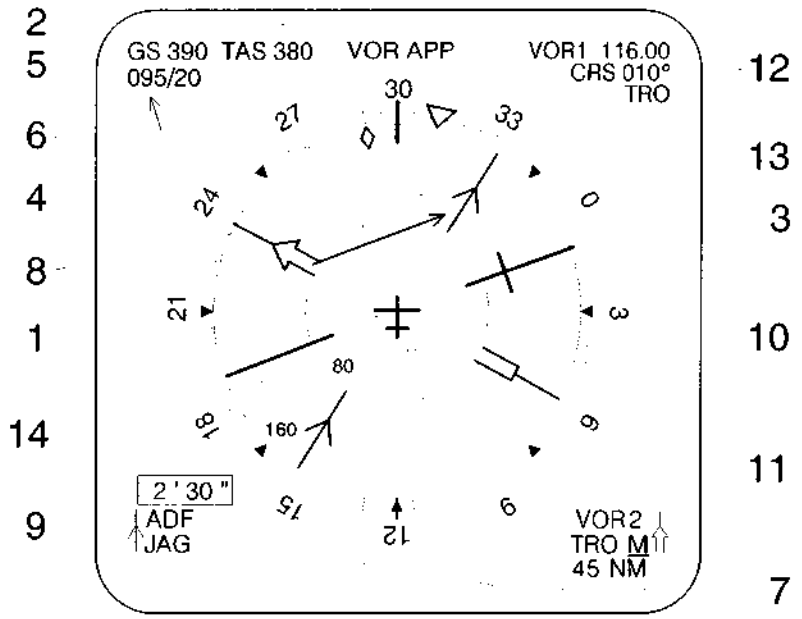
- ROSE VOR
- ROSE ILS
- ROSE NAV
- ARC
- PLAN

A *weather radar* image can be superimposed on the ND in all modes except PLAN.

NOTE: In case of avionic ventilation failure, the *weather radar* image is suppressed in order to limit DU power consumption and to prevent them from overheating.

1. Aircraft symbol (yellow): Fixed, pointing vertically up to the yellow lubberline.
2. Aircraft *heading*: The aircraft magnetic *heading* is given by a fixed yellow lubber line and a white moving compass rose. Fixed small white triangles are arranged at 45° intervals around the *circumference* line. "TRU" is displayed when true *heading* is displayed *instead* of magnetic *heading* (latitude above N 73° or S 60°).
3. Selected *heading* or *track* (blue): Displays the value indicated on the HDG TRK counter of the FCU.
4. Actual *track* (green)
5. Ground Speed / True Air Speed (green)
   Computed by ADIRS.
6. Wind direction and speed: Wind direction in numeric form with respect to true north and by green arrow with respect to magnetic north (only displayed when wind speed is >2 kt). In case of no wind or no wind information available, the *corresponding* numerical value is replaced by dashes (ADIRS computed)
7. Navaids: When on the EFIS control panel either ADF/OFF/VOR selector switch is set to ADF or VOR, the following characteristics of the *corresponding* navaid are displayed in white for VOR or in green for ADF on the onside ND:
   a) Type of navaid (ADF or VOR-1 on left side, ADF or VOR-2 on right side).
   b) Shape and color of *associated* bearing pointer
   c) Navaid identification.
   d) DME distance if VOR is selected and a co-located DME station is available.
   e) Mode of tuning.
   f) M (underlined and dimmed) for navaid manually tuned on the MCDU.
   g) R (underlined and dimmed) for navaid tune on an RMP.
   h) Nothing for navaid automatically tuned by the FMGC.

Figure 8: ROSE VOR Mode



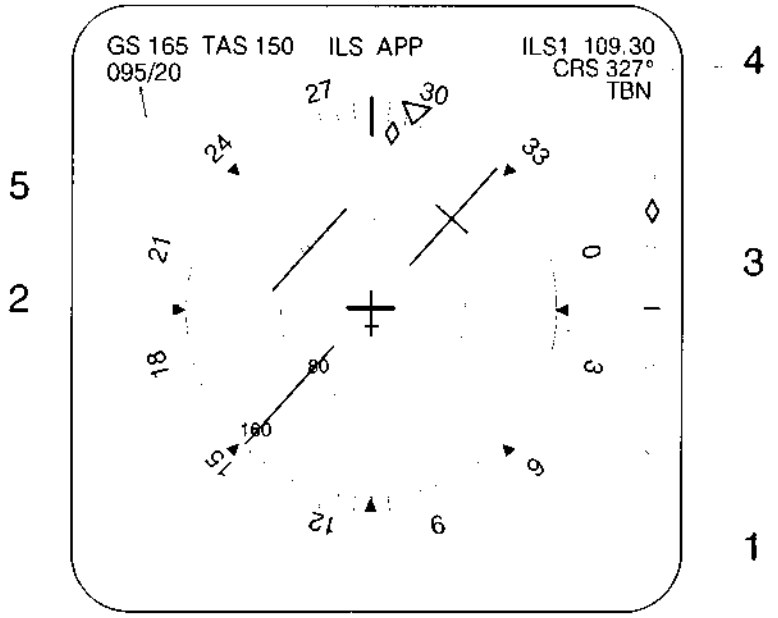
8. Bearing pointer:
   a) ADF in green
   b) VOR in white
   c) If the tuned navaid is not received, the *associated* bearing pointer is not displayed.
9. Chrono indication (white): Displays elapsed time when onside chrono is started.
10. VOR course pointer (blue): *Dagger* shaped symbol points to the selected VOR course. The VOR course is selected either by FMGC (auto tuned or manually) or through RMP in back up mode.

11. *Lateral deviation* bar (blue): Indicates the VOR *deviation* on a *lateral* scale. Each *dot* represents 5°. When *lateral deviation* is above 10°, the bar remains displayed on the outer *dot*. The arrow on the bar gives the TO/FROM indication.
12. VOR information (white): Selected VOR frequency, course and identification (decoded by the receiver) and the mode of tuning (M or R).
13. VOR APP message (green): Displayed when a VOR approach has been selected.
14. Range marks: The *range* scale value selected on the EFIS control panel (10 to 320 NM) governs the scale of the ND.

ROSE ILS Mode

1. ILS course pointer (*magenta*): *Dagger* shaped symbol points to the selected ILS course. The ILS is selected either by FMGC (auto tuned or manually) or through the RMP in back up mode. If no course has been entered, the value is defaulted to 360°.
2. *Localizer deviation* bar (*magenta*): Moves *laterally* on a scale with respect to the course pointer. Its scale *consists* of two *dots* on either side of zero *deviation*. Each *dot* represents a *deviation* of about 0.8°.
   In case of *excessive deviation* (1/4 *dot*) above 15 ft RH, the bar and scale will pulse, *provided* LOC TRK or LAND TRK mode is engaged.
3. *Glide slope deviation*: *Magenta* diamond moves on a vertical scale. The scale *consists* of two white *dots* on each side of the yellow reference line. Each *dot* represents a *deviation* of about 0.4°.
   In case of *excessive deviation* (1 *dot*) above 100 ft RH, the scale and diamond will flash, *provided* G/S TRK or LAND TRK mode is engaged.
4. Selected ILS information: Selected ILS frequency (*magenta*) course (blue) and identification (*magenta*).
5. ILS APP message (green): Displayed when an ILS approach has been selected.
   ILS-1 information is displayed on PFD-1 and ND-2.
   ILS-2 information is displayed on PFD-2 and ND-1.

Figure 9: Rose ILS Mode



Rose Nav Mode

See "Figure 11: Rose Nav Mode" on page 12

6. *Range* marks and values: Inner circle represents 1/4 of the selected *range*. *Heading* scale 1/2 of the selected *range*.
7. ILS course (*magenta*): Displayed when the ILS key on EFIS control panel is pressed, *provided* an ILS station has been selected.

Figure 10: NAVAIDS

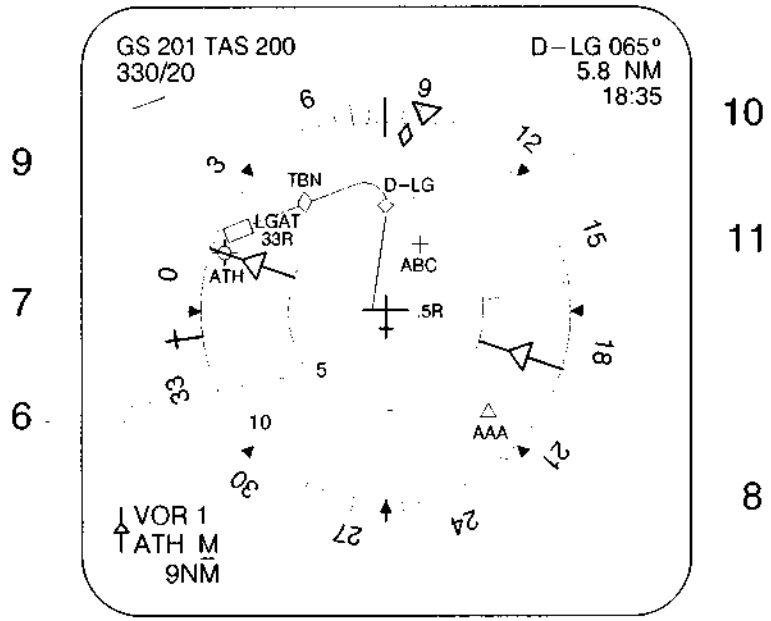
- ○ DME or TACAN
- + VOR
- ⊹ VOR/DME
- △ NDB

Navaids are displayed as follows:

- Green if it is a current waypoint of the actual flight plan.
- White if it is the TO waypoint.
- Blue when it is tuned for display either automatically by the FMGC or manually through the MCDU.
- *Magenta* when the navaid is not part of the flight. (selected for display with *option* keys on EFIS control panel).

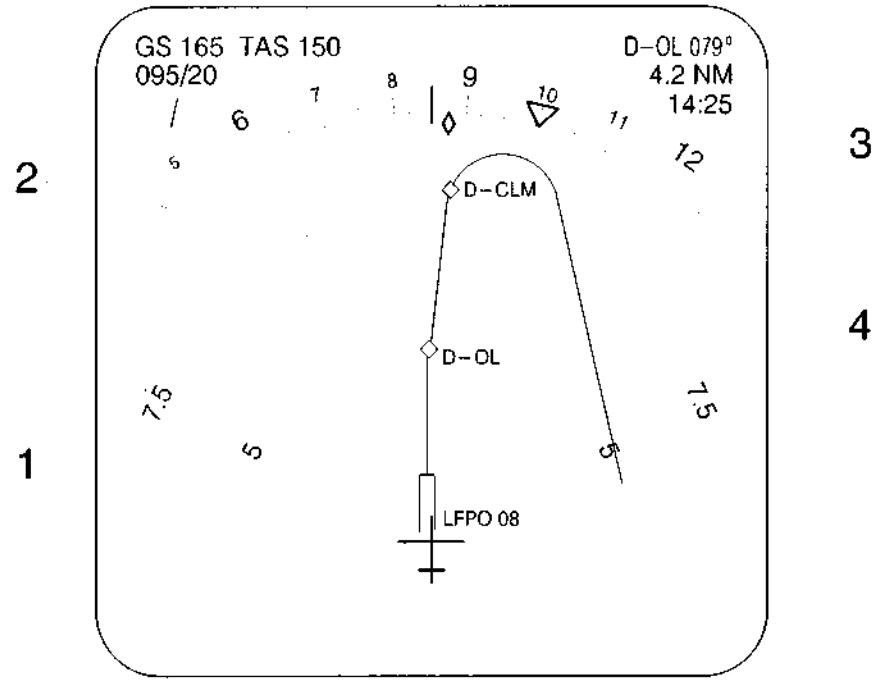
8. Flight plan (see ARC mode)
9. TO waypoint information (see ARC mode)
10. Cross *track* error: *Lateral deviation* in NM Left or Right from the active flight plan (related to great circle route).

Figure 11: Rose Nav Mode



ARC Mode

Figure 12: ARC Mode



ARC mode *provides* the same information as ROSE NAV mode, limited to the forward 90° sector.

1. *Range* marks and values: Inner circle represents 1/4 of the selected *range*. Second circle represents 1/2 of the selected *range*. Third circle represents 3/4 of the selected *range*. *Heading* scale represents selected *range*.
2. Flight plan: Various types of flight plan can be displayed, selectable through the MCDU.
   - Active flight plan: A continuous green line represents the flight plan actually followed by the aircraft when AFS NAV mode is engaged. Only the part of the flight plan which is ahead of the aircraft is displayed as well as the waypoints which are still to be overflown plus the waypoint from which the aircraft is coming.

SID and STAR, except the last WPT of the SID and the first WPT of the STAR are not displayed when ND *range* 160 or 320 is selected.
If the primary flight plan is not active, it is represented by a green *dotted* line.

- Missed approach and alternate flight plan:
  The missed approach is represented by a continuous blue line and the alternate flight plan route by a *dotted* blue line.
  They are displayed in ARC, ROSE NAV or PLAN mode when a missed approach waypoint and/or an alternate F–PLN waypoint are displayed on the onside MCDU.
- Secondary flight plan: Represented by a continuous white line. Active flight plan remains displayed.
- Temporary flight plan: The revised portion of the flight plan is represented by a yellow *dotted* line.
- Flight plan *capture*: When the aircraft is off the primary flight plan and flying towards it in HDG mode and NAV mode armed, the new active flight plan is displayed in a continuous green line *provided* the FMGC has computed the intercept path.
  The part of the flight plan which is before the interception point is drawn as a green *dotted* line.

3. Waypoints:
   - TO waypoint (white)
     Next waypoint to be overflown. The following *associated* data is displayed in the right upper corner:
     - Waypoint identification (white) / Desired *track* (green).
     - Distance to go (green).
     - Estimated time of arrival (green).

4. Airports

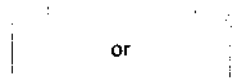
Airport included in flight plan

LSGG	Airport which are part of the flight plan, but without specified runway are represented in white.
LSGG 23	Runway specified. Oriented runway symbol drawn to scale (paved length) if 10, 20 or 40 NM display range is selected.

Airport optional information
The airports which are not displayed as part of the flight plan may be called for display by pushing the ARPT pushbutton on the EFIS control panel.
The are represented by a star and the identification in magenta.

Procedure turns and holding patterns
Displayed only when they are part of the flight plan.
When 160 or 320 NM range is displayed, a white arrow indicates the turn direction.
For other ranges, if the procedure turn or the holding pattern is in the next or the active leg, the full circuit or pattern is displayed.

Range 160 or 320 NM



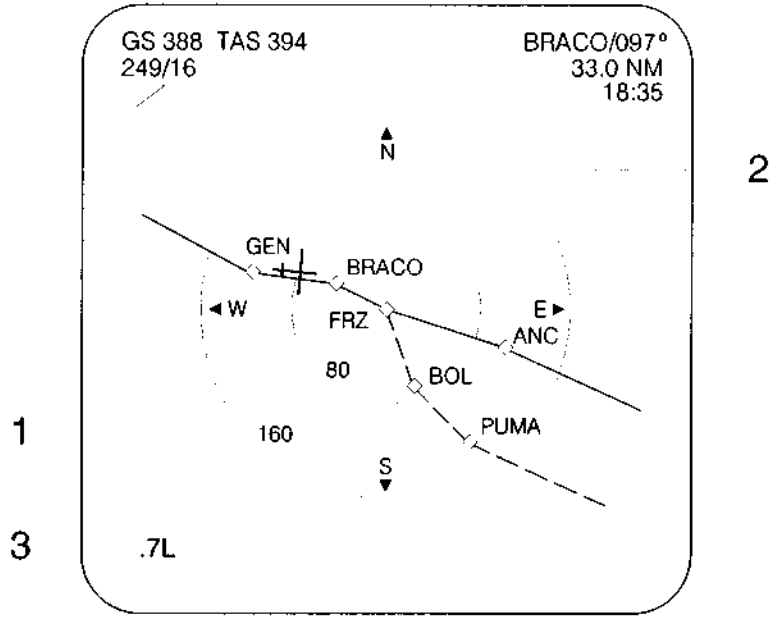
Range 10, 20, 40 or 80 NM

PLAN mode

Displays statistically the F-PLN legs on a true north oriented map. The map is centered on a map reference point which is chosen by the pilot on his MCDU using slew keys. The scale of the map is chosen by the *range* selector (the diameter of the outer circle corresponds to the selected *range*).

Navaids data and bearing pointers are not available in this mode.

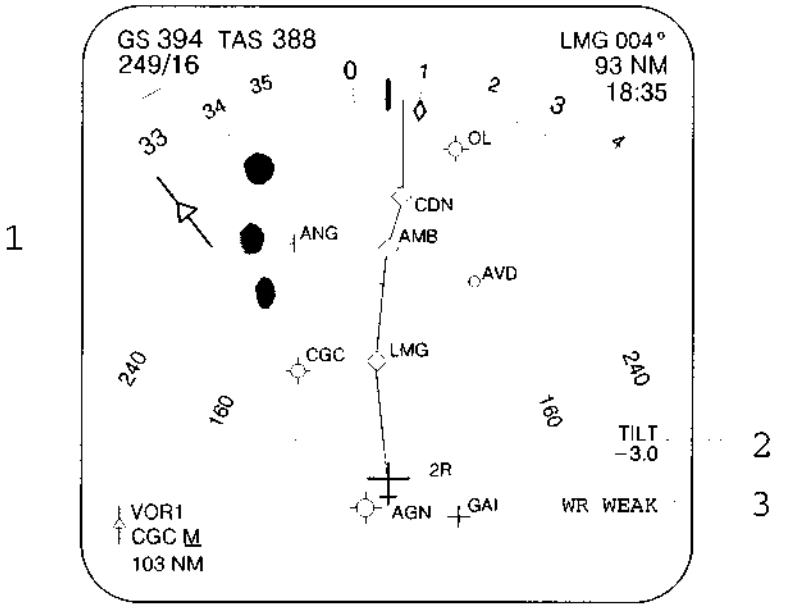
Figure 13: "Plan Mode



1. Aircraft symbol (yellow): The orientation of the aircraft symbol indicates the true *track*. Its position represents the actual aircraft position in respect to the flight plan.
2. Map reference point: Waypoint displayed on the second line of the flight plan displayed on the MCDU F-PLN page. It can be either the active TO waypoint or any other waypoint of the flight plan.
3. Offset annunciation: Indicates cross *track deviation* to the left (L) or right (R) of the flight plan route in NM.

Weather Radar Display

Figure 14: *Weather Radar* **Display**



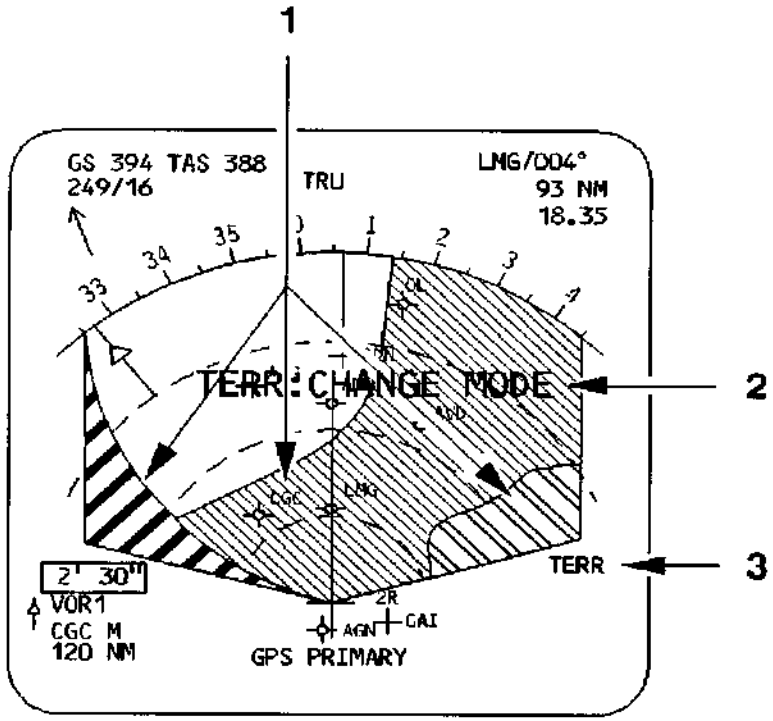
1. *Weather radar* image: Displayed when the radar is selected on and the ND is not in PLAN mode.
   The echoes are displayed in green, yellow, red and *magenta* according to the precipitation rates. The echo refresh rate depends on the selected *range*.
2. Tilt angle / *gain* mode: Shows antenna tilt angle (angle between horizon and radar beam axis) in degrees and quarter of a degree (blue). When selected, the MAN *gain* mode is displayed in green.
3. Failure messages: Following failure message can be displayed:
   - WR RT (red) radar transceiver failure
   - WR ANT (red) antenna failure
   - WR CTL (red) control panel failure
   - WR RNG (red) *range* error
   - WR WEAK (*amber*) calibration failure
   - WR ATT (*amber*) *attitude* information failure

– WR STAB (*amber*) antenna stabilization failure

In case of red failure messages, no radar image is displayed. If the message is in *amber*, the image is not affected.

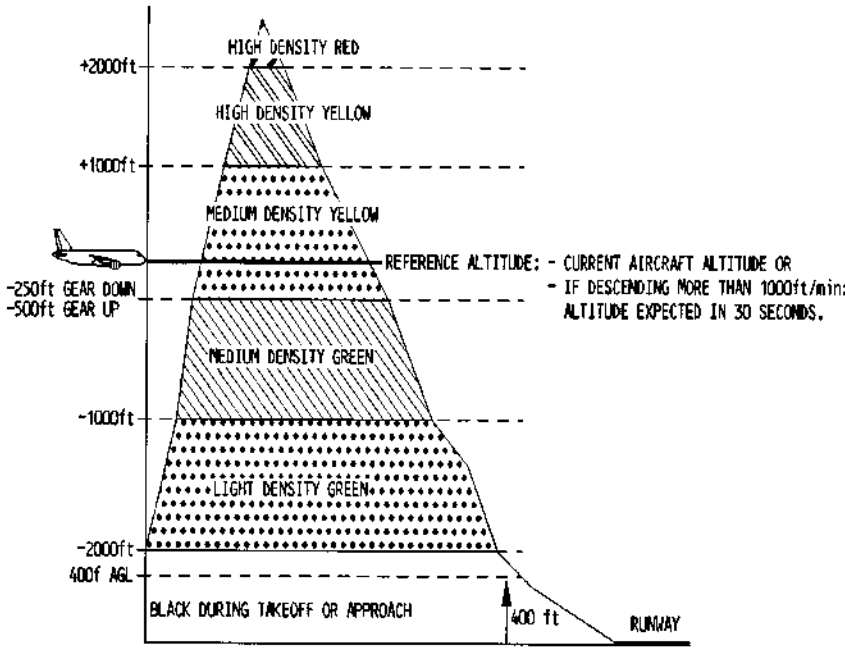
EGPWS

Figure 15: EGPWS Display



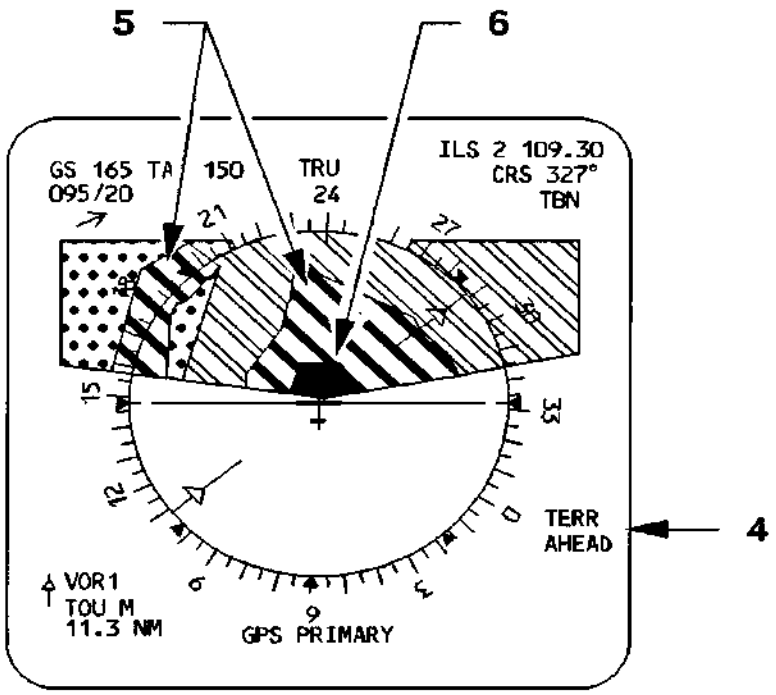
1. EGPWS terrain picture: The ND presents the EGPWS terrain picture when the TERR ON ND switch is selected ON and the ND is not in PLAN mode. The terrain picture replaces the *weather radar* image.
   The terrain appears in different colours and densities according to its relative height: see "Figure 16: Colour Code of the EGPWS" on page 15.

Figure 16: Colour Code of the EGPWS



2. TERR: CHANGE MODE indication: Displayed in red (or *amber*) in case of Terrain *Awareness* Display (TAD) warning (or caution) *alert*, if the current selected display mode is PLAN.
3. TERR indication: To *differentiate* the terrain from the weather display, the *weather radar* TILT is replaced by a blue TERR, and the terrain display sweeps from the center outward to both sides of the ND.
4. Warning and caution messages:
   – TERR AHEAD (*amber*) for a caution
   – TERR AHEAD (red) for a warning
   When *triggered*, these messages are flashing 9 seconds, then remain steady until the caution or warning *alert* condition disappears.
   – TERR RNG (red) for a *RANGE* error warning
   – TERR TST (*amber*) appears during the EGPWS test when the terrain pattern, is displayed and there is no failure.

Figure 17: EGPWS Display



5. Terrain caution *alert*: This *alert* is generated when a conflict exists between the terrain caution envelope ahead the aircraft and obstacles stored in the database. The area of conflict is displayed in solid yellow.
6. Terrain warning *alert*: This *alert* is generated when a conflict exists between the terrain warning envelope ahead the aircraft and obstacles stored in the database. The area of conflict is displayed in solid red.

NOTE: When an *alert* is generated (either caution or warning) and the TERR ON ND is not selected, the terrain is automatically displayed and the ON light of the TERR ON ND push button switch comes on.

CAUTION:

a) The relative height of the aircraft is computed using the captain *baro setting*. Thus the Terrain *Awareness* Display (TAD) does not protect against *baro setting* errors.

b) The Terrain *Awareness* Display and Terrain Clearance Floor (TCF) functions operate using the FMS 1 position. Thus, the system does not protect against FMS 1 position error.

ECAM

The ECAM (Electronic Centralized Aircraft Monitor) *consists* of 2 DU's (upper and lower), 2 Flight Warning Computers (FWC), 2 System Data *Acquisition* Concentrators (SDAC), master warning and master caution lights for each pilot and the ECAM control panel.

The ECAM presents following data on an Engine/Warning Display (E/WD) and a System Display (SD):

- Engine primary indications, fuel quantity indication, flaps and slats position indication.
- Warning and caution *alerts* or memo messages.
- A/C systems synoptic diagrams or status messages.
- Permanent flight data.

The DU's are driven by independent ECAM channels *contained* in the DMC's.

System Data Acquisition Concentrator (SDAC)

Two identical units *acquire* data required by the 3 DMC's, for the display of system pages and engine parameters and by the FWC's for generation of ECAM messages and *aural alerts*.

Flight Warning Computer (FWC)

Two identical computers *acquire* data for the generation of *alert* messages, memo informations, *aural alerts*, synthetic voice messages and flight phases computation:

- Directly from A/C sensors or systems to generate red warnings.
- Through the SDAC's for *amber* cautions.

The *alert* messages *elaborated* by the FWC's are displayed on the ECAM DU's. A database in the FWC's enables to store a list of warnings, cautions and procedures which are affected by Operations Engineering Bulletins (OEB). Items affected by such a bulletin cause the display of the message "REFER TO QRH PROC" on the EWD and/or SD.

Additional FWC functions are:

- Radio *altitude* call out.
- Decision height call out.
- Landing distance and landing speed *increments* computation.

Aural alert and voice messages are *emitted* through the cockpit loudspeakers, even when they are switched off.

Color code

A color code is used on both ECAM DU's according to the importance of the failure or the indication.

- RED: For configuration or failure needing immediate action.
- *AMBER*: For configuration or failure needing *awareness* but not immediate action.
- GREEN: For normal long term operation.
- WHITE: For titles and remarks used to guide during procedures.
- BLUE: For actions to be carried out or limitations.
- *MAGENTA*: For particular messages (e.g. for inhibition messages).

Priority rules

Three *levels* have been defined for warnings / cautions:

A *level* 3 warning has priority over *level* 2 caution which has priority over *level* 1 caution.

In a same *level*, an order of priority has been defined in the FWC.

Types of failure

Independent failure: Failures which affect an isolated item of equipment or system without influence on others in the aircraft.

Primary failure: Failures of an item of equipment or system causing *loss* of others in the aircraft.

Secondary failure: *Loss* of an item of equipment or system resulting from a primary failure

ECAM Warnings and Cautions

Table 1: ECAM Warning / Caution classification

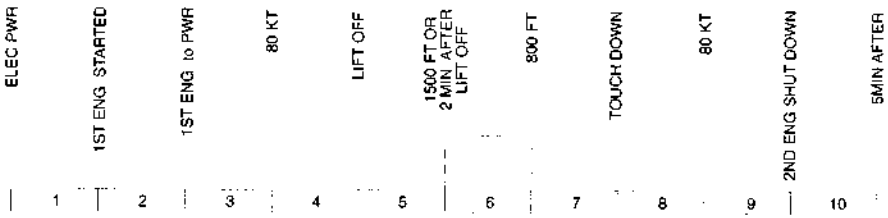
	Level	**Signification**	*Aural*	**Visual**
Failure Mode	*Level* **3**	**Red warning** Situation needing immediate action: • Aircraft in dangerous configuration or limit flight conditions (e.g. *stall*, overspeed). • System failure altering flight safety (e.g. eng. fire, excess cabin alt)	Continuous Repetitive Chime (CRC) or specific sound or synthetic voice.	• Master Warn light red flashing or specific red light. • Warning message (red) on E/WD. • Automatic call of the *relevant* system page on SD*.
	Level **2**	*Amber* **caution** Abnormal situation needing immediate crew *awareness* but no immediate action. System failure having no direct consequence on flight safety (e.g. BLUE HYD LO PR).	Single Chime (SC).	• MASTER CAUT light *amber* steady. • Caution message (*amber*) on E/WD. • Automatic call of *relevant* system page on SD*.
	Level **1**	*Amber* **caution** Requires crew monitoring. Failures leading to a *loss* of redundancy or system degradation.	None	• Caution message (*amber*) on E/WD generally without procedure.
Information	*Advisory*	System parameters monitoring	None	• Automatic call of the *relevant* system page on the SD. The affected parameter pulses green.
	Memo	Information Recalls normal or automatic selection of functions which are temporarily used.	None	• Green, *amber* or *magenta* message on E/WD:

* Some warnings do not automatically call up a system page.

Flight phases

The FWC computes ten flight phases:

Figure 18: Flight Phases



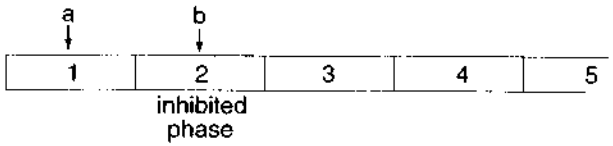
In order to avoid *disturbing alerts* especially during high workload phases like take-off or landing, warning/caution inhibitions related to flight phases are performed.

When those inhibitions are in effect, following *magenta* MEMO messages will be displayed: T.O. INHIBIT, LDG INHIBIT

NOTE: These flight phases are different and independent from the ones used by the FMGC.

Flight Phase Inhibition

Figure 19: Flight Phase Inhibition



Effect on E/WD:

a) If a failure occurs before the *inhibited* flight phase, the warning will come on immediately and remain displayed as long as it is present, even if a inhibition phase follows.

b) If the failure occurs during an inhibition flight phase, the *alert* will come on only when the inhibition phase is over and will remain displayed as long as the failure is present.

Configuration warnings

The FWC's *trigger* following warnings/cautions when the aircraft is not in take-off configuration when the T.O. CONFIG push button is pressed (ECAM control panel) or when take-off power is applied:

Table 2: Configuration Warnings

SLATS/FLAPS NOT IN TO *RANGE* (red)	**T.O. Config Test**	**T.O. Power applied**
PITCH TRIM NOT IN TO *RANGE* (red)		
RUD TRIM NOT IN TO *RANGE* (red)		
SPEED BRAKES NOT RETRACTED (red)		
SIDESTICK FAULT (red)		
BRAKES HOT (*amber*)		
DOOR NOT CLOSED (*amber*)		
PARK BRAKE ON (red)		
FLEX TEMP NOT SET (*amber*)		

Sounds Definition

Table 3:

WARNING SIGNALS	CONDITION	DURATION	SILENCING *
Continuous Repetitive Chime	Red warnings	Permanent	Depress ** MASTER WARNING light.
Single Chime	*Amber* caution	1/2 sec.	NIL
Cavalry charge	A/P disconnection by TAKE OVER p/b	1.5 sec.	Second push on TAKE OVER p/b.
	A/P disconnection due to failure	Permanent	Depress MASTER WARNING light or TAKE OVER p/b.
Click	Landing capability change	1/2 sec. (3 pulses)	NIL
Cricket +*STALL* (synthetic voice)	*Stall*	Permanent	NIL
Buzzer* **	Cabin call	3 sec.	NIL
	Emergency cabin call	3 sec. repeated 3 times	NIL ***
	Mechanic call	As long as p/b depressed.	NIL
Continuous Buzzer * **	SELCAL call	Permanent	Depress RESET key on ACP.
C Chord	*Altitude alert*	1.5 sec. or permanent	Select new *altitude* or press MASTER WARNING light.
Auto call out 300, 100, 50,40, 30, 20, 10 MINIMUMS (synthetic voice)	Height announcement below 400 ft	Permanent	NIL
Ground Proximity Warning (synthetic voice)	See GPWS	Permanent	NIL

Table 3:

WARNING SIGNALS	CONDITION	DURATION	SILENCING *
WINDSHEAR **(synthetic voice)**	*Windshear* Repeated	3 times	NIL
PRIORITY LEFT PRIORITY RIGHT (synthetic voice)	A/P take over p/b	1 sec.	NIL
RETARD (synthetic voice)	Thrust lever not in idle position for landing	Permanent	Retard thrust levers.
TCAS (synthetic voice)	See TCAS	Permanent	NIL
SPEED SPEED SPEED (synthetic voice)	Current thrust is not sufficient to recover a positive flight through pitch control.	Every 5 seconds until thrust is increased.	Thrust lever(s)
DUAL INPUT (synthetic voice)	Both sidesticks are moved *simultaneously*	Every 5 seconds	One sidestick deactivated

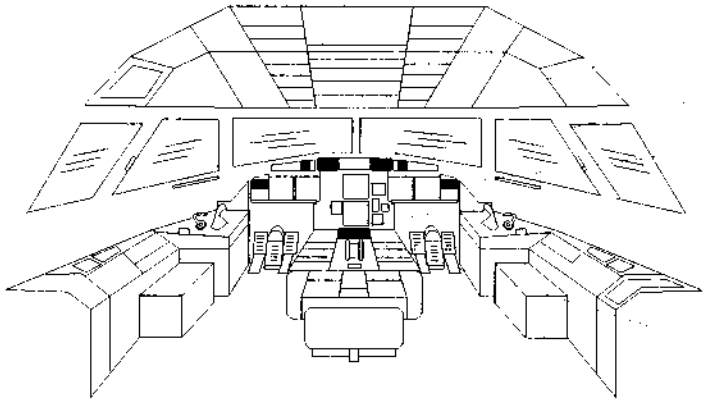
* All *aural* warnings may be cancelled by depressing the EMER CANC key on the ECAM control panel.

** Except for OVERSPEED or L/G NOT DOWN warnings.

*** May be cancelled by depressing MASTER CAUT push button.

Controls

Figure 20: Cockpit Control Panels related to the EIS System



C

D

B

A

E

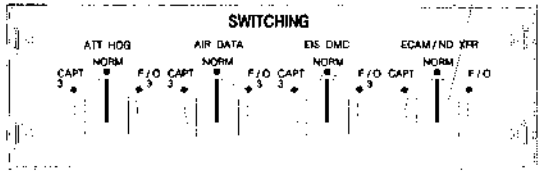
ECAM / ND Selector

Allows to transfer SD to either Captain's or Copilot's ND. In case of dual failure (E/WD and SD), the selector allows to transfer the E/WD to either ND.

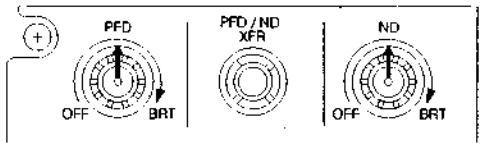
EIS DMC Selector

NORM	DMC-1 supplies data to PFD-1, ND-1 and upper ECAM DU. DMC-2 supplies data to PFD-2, ND-2 and lower ECAM DU.
CAPT 3	DMC-3 replaces DMC-1.
F/O 3	DMC-3 replaces DMC-2.

A EIS Switching Panel



B EFIS Display Controls



PFD / ND XFR Rushbutton

Allows to manually transfer the PFD to the NDU and vice versa. In case of PFD failure, the PFD is automatically transferred to the NDU.

ND Brightness Control Knob

The outer knob controls the brightness of both the weather radar image and the EGPWS(if installed) terrain display.

The inner knob controls the general brightness of the ND symbology. Full counterclockwise rotation switches the ND off.

PFD Brightness Control Knob

Full counterclockwise rotation switches the PFD off. In this case the PFD image is automatically displayed on the NDU. The ND may be recovered by means of the PFD / ND XFR pushbutton.

Figure 21: EFIS Control Panel and Master Warning, Master Caution Lights

Baro Reference Display Window
Range 745 HP to 1050 HP.

Baro Reference Selector

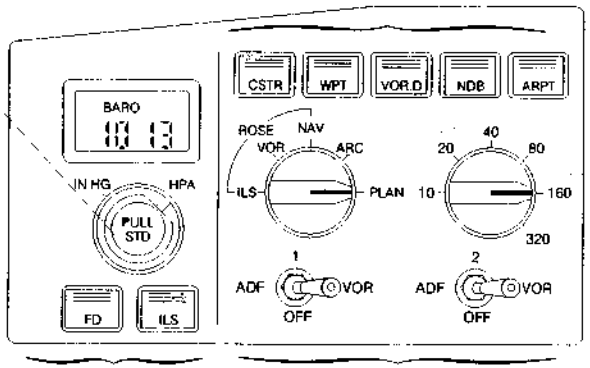
Outer knob : Rotating it selects either HPA (Hecto Pascal) or IN HG (inches of mercury) display.

Inner knob : Adjusts the QNH value either in HPA or IN HG.
Pulling knob selects standard baro reference setting. STD will be displayed on PFD. Rotating the knob in STD mode has no effect.
Pushing the knob from STD position activates the QNH bar setting again.

FD Key
Selects/deselects FD bars display on PFD or Flight Path Director symbol display, when TRK FPA reference is se lected.
The bars on the key are illuminated when FD/FPD display is selected ON.

ILS Key
Selects/deselects LOC and GS display on the PFD. The deviation symbols are displayed when a valid ILS signal is received.
The bars on the key are illuminated when ILS display is selected ON.

C EFIS Control Panel



Optional Data Display Keys (5)
Selects/deselects display of optional data in addition to the permanently displayed flight plan data, when in PLAN, ARC or ROSE NAV mode. Only one key can be active at a time.
The bars on the key are illuminated when function is ac tive.

Display Mode Selector
Selects desired navigation display on onside ND.

Range Selector
Selects desired viewing range on onside ND:
NOTE: In case of mode or range data failure, the default selection will be ROSE NAV mode and 80 NM range.

ADF / VOR Selectors (2)
Selects display of the VOR / ADF bearing pointers and DME distance on onside ND.

D

MASTER WARN Lights (2)
Flash red for level 3 warnings, accompanied by a continuous repetitive chime, specific sound or synthetic voice.
The light will extinguish when the warning situation no longer exists or when:
- The MASTER WARN light is pushed (except for overspeed and STALL warning).
- The ECAM panel EMER CANC key is pushed.
- The ECAM panel CLR key is pushed (except for overspeed and STALL warning).

The aural warning stops when:
- The MASTER WARN light is pushed (except for overspeed and STALL warning).
- The ECAM panel EMER CANC key is pushed.

MASTER CAUT Lights (2)
Illuminate steady amber for level 2 cautions, accompanied by a single chime.
The light will extinguish when the warning situation no longer exists or when the MASTER CAUT, the CLR key or the EMER CANC key is pushed.

Figure 22: ECAM Control Panel

System Page Keys (11)

Depress to select the desired system page on the SD. Key illuminates after manual selection or when an advisory is detected.
Depressing the illuminated key a second time recalls the system page related to the present flight phase or the current warning.
In mono ECAM display, system pages may be displayed by pressing and holding the system page key .
- In case of an advisory condition, the key light will flash (no automatic system page display).
- In case of of a warning condition, there will be no illuminated key light nor automatic system page display.

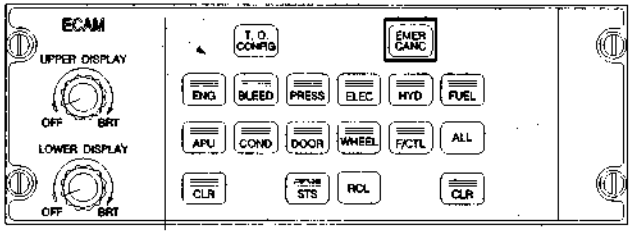
STS Key

When depressed the STATUS page appears on the SD. Illuminates as long as the STS page is displayed. If there is no status message present, NORMAL will be displayed for 5 seconds.
The STATUS page may be cleared by the CLR key or by depressing the STS key a second time.
In mono ECAM display:
- The STATUS is only displayed when the STS key is pressed and hold. The next status page, if any will be displayed by releasing the key and pressing it again.
- Holding the STS key depressed, displays the status page for 30 seconds, then the E/WD will automatically appear again.

Display Brightness Control Knobs (2)

Full counterclockwise rotation switches the associated DU off.
When the upper display is turned off , the E/WD image appears automatically on the lower DU .
Manual brightness control is combined with automatic adjustment of the display brightness according ambient light conditions.

E ECAM Control Panel



RCL Key (2)

Depress to recall warning/caution messages and the status page previously cleared by the CLR key or by the flight phase related automatic inhibition.
If no warning/caution is present, NORMAL will be displayed for 5 seconds.
When depressed more than 3 seconds, the caution messages which have been cancelled by the EMER CANC key are recalled.

TO CONFIG Key

When depressed, TO power application is simulated.
If the aircraft is not in TO configuration, a warning will be triggered. If the configuration is correct, the message T.O. CONFIG NORMAL is displayed.

EMER CANC Key (guarded)
- Warnings (level 3)
  Depressing the key cancels the present aural warning and extinguish MASTER WARNING lights.
  The ECAM message display is not affected.
- Cautions (level 2 and level 1)
  Depressing the key cancels any present caution alert (MC lights, ECAM message) for the rest of the flight.
  The STATUS page is automatically selected, displaying following message:
  CANCELLED CAUTION
  .....(title of failure which is inhibited)

The inhibition is automatically cancelled when flight phase 1 is initiated or manually when RCL key is depressed for more than 3 seconds.

ALL Key

When depressed and held, all system pages appear successively at 1 seconds interval as long as the key is depressed. When released the presently displayed page will remain displayed.

CLR Key

Illuminates as long as a warning/caution message or a status message is displayed.
Depressing the CLR key when illuminated, will change the ECAM display.

Embraer 145 EIS Presentation

Introduction

Similar as on the other aircraft types the Electronic Instrument System on the Embraer 145 is divided into the following systems

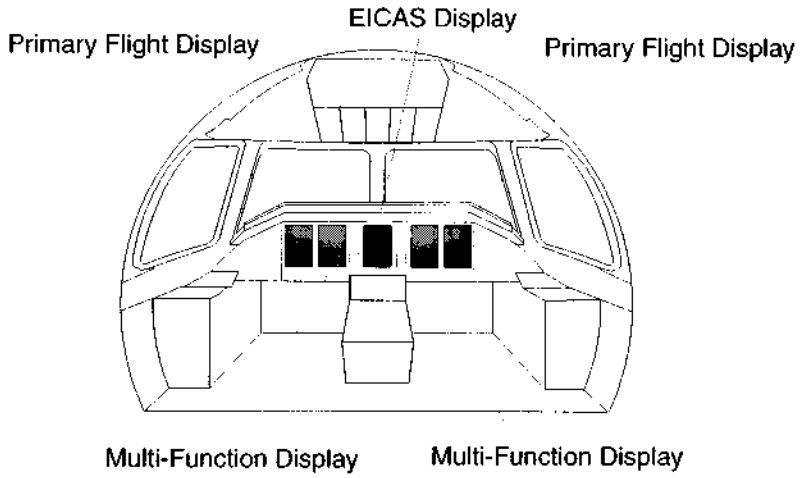
- EFIS (Electronic Flight Instrument System)
  providing all primary flight and navigation informations to the crew, and
- EICAS (Engine Indicating and Crew *alerting* System)
  visualisation of the information concerning normal and abnormal *state* of the Engines and the aircraft systems

The output of these systems is presented in the cockpit on five Display Units as shown in Figure 23 and via the Loudspeakers.

The five Displays are:

- Primary Flight Display (PFD) on DU#1, pilot side
- Multi-Function Display (MFD) on DU#2, pilot side
- EICAS Display on DU#3, Center Main Instrument Panel
- Multi-Function Display (MFD) on DU#4, copilot side
- Primary Flight Display (PFD) on DU#5, copilot side

Figure 23: Cockpit Arrangement of the Display Units



Display Description

- Primary Flight Display - PFD (DU's 1 AND 5)
  The PFD is the primary flight reference for the flight crew members.
  The PFD has references for *airspeed*, *altitude*, *attitude*, *heading*, navigation, and radio aids.
- Multi-function Display - MFD (DU's 2 AND 4)
  The MFD is used as navigation display in normal operation, but it can be either a PFD or EICAS display.
  Through the MFD, you can set up to five system pages, one electronic checklist page, one TCAS page and one displayable maintenance messages page.
  Only the left MFD (DU-2) is also used to show the displayable maintenance messages.
- EICAS Display (DU-3)
  The EICAS display shows engine parameter, fuel system, flight control system, ECS, APU, and crew *alerting* systems.

Electronic Instrument System Architecture

The Integrated Computers system is the primary component of the whole avionics system. The Integrated Computers system together with controllers and sensors form the Primus-1000 (P-1000) Honeywell System.

These computers are called Integrated Computers because there are different parts with different functions, integrated into one box.

Within the Electronic Instrument System, the Integrated Computers perform symbol generator function for the EFIS, symbol generator function as well as *processing* of the different warnings and messages for the EICAS.

Other parts of the Integrated Computer are used for flight director and autopilot functions.

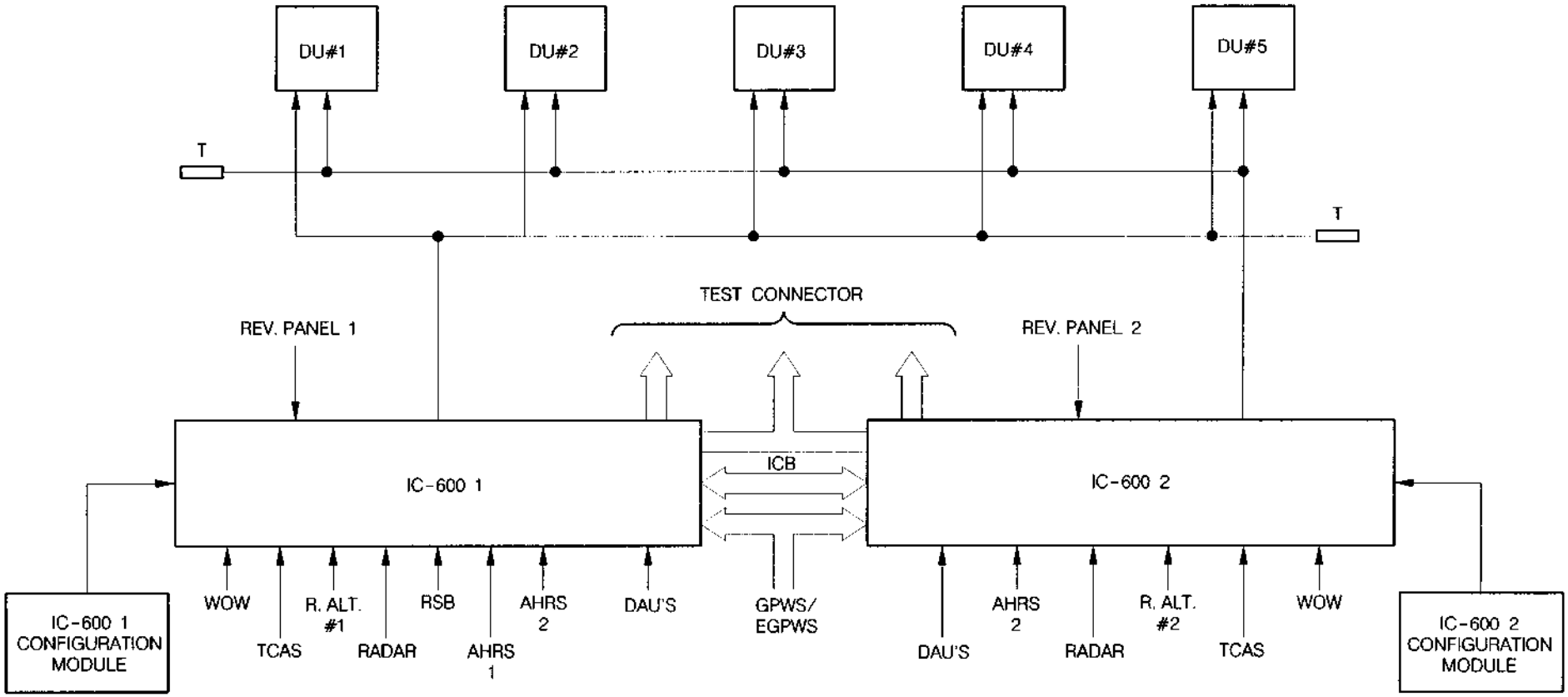
Between the Integrated Computers, there is a cross-talk bus called Intercommunication Control Bus (ICB), which is used for comparison purposes.

The IC # 1 normally *provides* images to DU's 1/2/3, while the IC # 2 *provides* images to DU's 4/5.

Figure 24 shows the architecture of the system.

A detailed description of the interfaces to the different aircraft systems shown in the block diagram will follow in the paragraphs on EFIS and EICAS.

Figure 24: Electronic Instrument System Architecture



IC 600	Integrated Computer
WOW	*Weight* on Wheel Sensing
TCAS	Traffic Collision Avoidance System
R. Alt	Radio Altimeter
Radar	*Weather Radar*
RSB	Radio System Bus
AHRS	*Attitude Heading* Reference System
DAU	Data *Acquisition* Unit
EGPWS	*Enhanced* Ground Proximity Warning System

Electronic Flight Instrument System (EFIS)

The Electronic Flight Instrument System (EFIS) *provides accurate* flight indications through its displays.

This system uses four of large 8" x 7" CRT displays:

- Two Primary Flight Displays (PFD's). Display Units (DU's 1 and 5).
- Two Multi-Function Displays (MFD's). Display Units (DU's 2 and 4).

These DU's receive image information from the two integrated computers (IC-600 1 and 2).

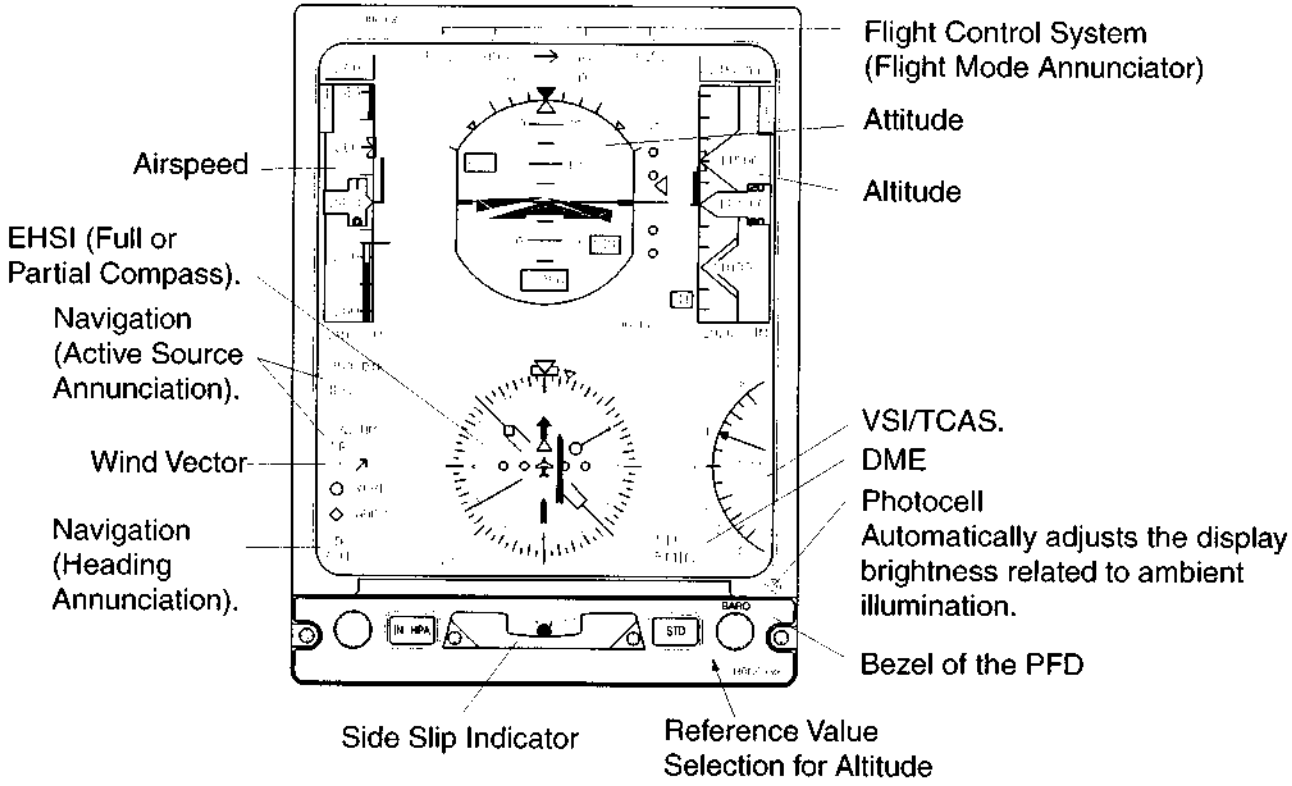
Both the Primary Flight Display (PFD) and the Multi-Function Display (MFD) are divided into two parts:

- Display Unit (DU).
- *Bezel.*

The DU's have the same part number, while the *bezels* have different part numbers.

On the Primary Flight Display, the most important parameters for the pilots to control the aircraft are displayed as shown on Figure 25.

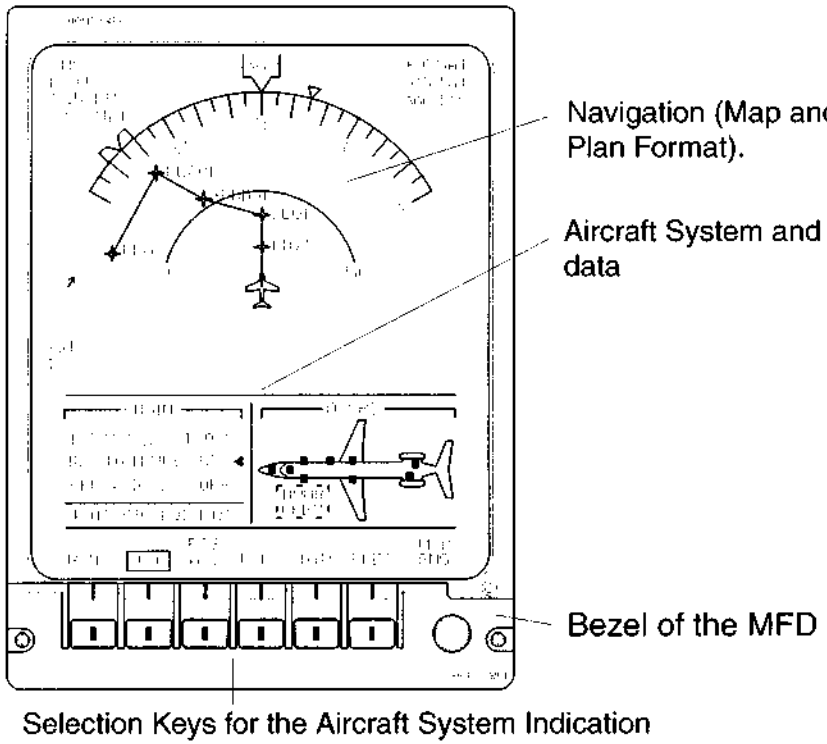
Figure 25: Description of the Primary Flight Display



The Multifunction Display is divided into two parts:

- On the upper part, navigation information is displayed which enables the pilots to navigate the aircraft.
- On the lower part of the display, information about aircraft systems and engines is displayed. In case of failure of a system, EICAS information such as electronic checklist can also be displayed.

Figure 26: Example of the Multifunction Display



Components of the EFIS

The description of the Components and interfaces that follows, will give you an overview of the EFIS System and will help you to understand the detailed diagram as shown on

The integrated display system of the aircraft has a 4-tube configuration.

This configuration includes:

- Two Primary Flight Displays (PFD's) DU-870
- Two PFD *bezels* BL-870
- Two Multi-Function Displays (MFD's) DU-870
- Two MFD *bezels* - BL-871
- Two Integrated Computers - IC-600
- Two Display Controllers - DC-550
- Two EFIS reversionary panels
- A Flight *Guidance* Controller (GC-550)
- Interface signals

Display Unit

The DU-870 is the electronic display used to supply the PFD EICAS and MFD information in the Primus-1000 system.

Each DU-870 is a CRT with three video guns (red, blue, and green) and with stroke and raster scanning techniques.

Bezel

Different *bezels* are attached to the front part of the DU-870 to make them either:

PFD, MFD, or EICAS displays.

Integrated Computer

The IC-600 is the primary component of all the avionics system. The IC-600 has:

- Symbol generator function
- EICAS function
- Flight director function
- Autopilot function

Each IC-600 gives three distinctive images: for PFD's, MFD's, and EICAS display.

Display Controller

The DC-550 controls:

- Navigation and *weather radar* display selection.
- Set bearing pointer sources.
- Self-test function selection.

It also moves analog signals from the GC-550 and *bezels*. Then, the DC-550 digitizes the signals to transmit them to the IC-600.

The DC-550 activates the self-test function for several systems.

Reversionary Panel

This panel supplies each flight crew member with cross-side information for:

- Symbol generator
- ADC
- AHRS (or IRS)
- Reversion of EFIS components

Flight Guidance Controller

This unit is responsible for the selection of:

- Vertical speed *target*
- Indicated *airspeed* or mach *target*
- Preset *altitude*
- Course
- *Heading*
- "Cross" bar or "V" bar

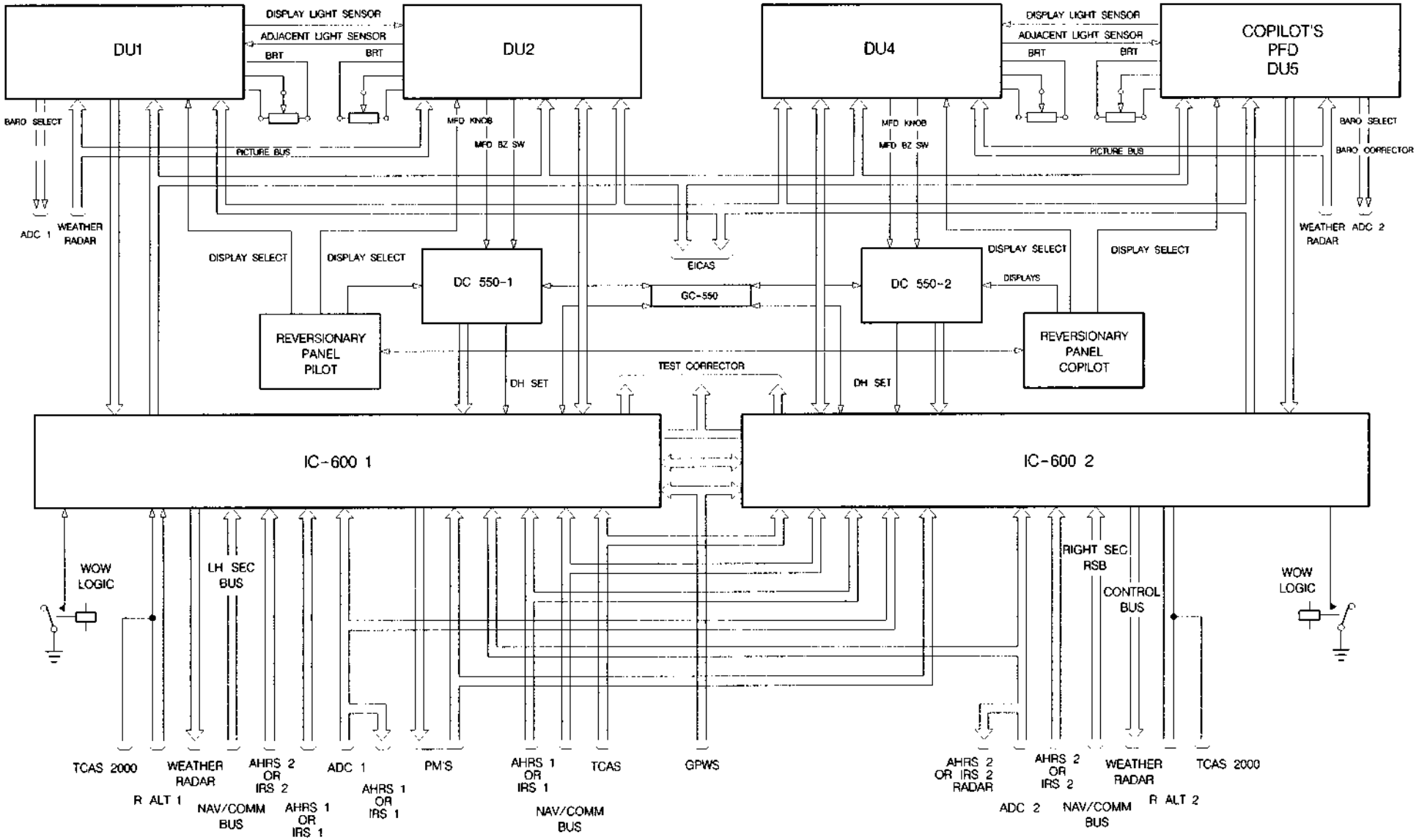
Interfaces to other Aircraft Systems

- AHRS (or IRS)
  The IC-600 receives information from the two AHRS sources (or from the two IRS sources). Each AHRS (or IRS) sends data separately to each IC-600. This makes sure that a single failure does not impair the data path from AHRS (or IRS) sources.
- PRIMUS II RADIO
  The Primus II radio system supplies navigation data to the IC-600 through the Radio System Bus (RSB). This bus transmits short *range* navigation data (VOR bearings, ILS approach data, marker beacons, DME and ADF). The RSB has:
  - Two secondary buses
  - One primary bus

The secondary buses receive information from the radio system, N.1 or N.2 separately, while the primary bus receives information from the two radio systems. The IC-600 1 receives data from a secondary bus and the primary bus. The IC-600 2 receives data from a secondary bus and the primary bus.

- MICRO AIR DATA COMPUTER - MADC (AZ-840)
  Each IC-600 receives data from the two MADC's. These data are:
  - Pressure *altitude*
  - *Baro altitude*
  - Mach
  - CAS
  - VMO
  - TAS
  - TAT
  - *Altitude*
  - SAT
  - *Baro* correction in HPA or In.Hg.
- RADIO ALTIMETER (AA-300)
  This system supplies analog radio *altitude* signals between - 20 ft to 2,500 ft.
- LONG *RANGE* NAV (FMS *OPTION*)
  The Primus-1000 receives LRN data for waypoint and navigation data. This data *permits* the IC-600 to show:
  - Flight plan waypoints
  - Waypoint bearing
  - Desired *track*
  - Cross *track deviation*
  - Waypoint distance
  - Time to waypoint.
- TCAS COMPUTER
  The IC-600 receives a vertical *deviation* signal for display on the PFD.
- DATA *ACQUISITION* UNIT (DAU)
  The IC-600 receives data from the DAU's to display information relative to engine, systems, and other aircraft sensors.
- *WEIGHT* ON WHEELS (WOW LOGIC)
  Both IC-600 receives WOW logic from an aircraft-mounted switch. This signal is grounded when the aircraft is on the ground and not when it is in flight. The WOW logic is internally *processed* in the Landing Gear Electronic Unit (LGEU).

Figure 27: Detailed EFIS Block Diagram



EICAS

The Engine Indication and Crew *Alerting* System (EICAS) receives and *processes* signals from engine and system sensors.

The architecture of the system (EICAS) starts from the engine and system sensors. In this system there are two Data *Acquisition* Units (DAU's) which collect data from systems of all parts of the aircraft.

The DAU 1 collects the data from the sources located in the front part of the aircraft and engine # 1. The DAU 2 collects the data from the sources located in the rear part of the aircraft and engine # 2.

Either message or indication goes through DAU, where the analog signal is changed to a digital one, and sent to the integrated computer (IC-600).

This system also uses reversion switches for display units, data *acquisition* unit, and symbol generators.

The Data *Acquisition* Unit (DAU) receives analog and digital signals for using them in the EICAS.

The following aircraft systems are interfacing with the EICAS for indication and alert functions:

- Power plant
- Landing gear
- Flaps
- Spoilers
- Pressurization
- APU
- Trims
- Brakes
- Hydraulics
- Electrical
- Fuel
- Ice/rain protection
- Oxygen
- Doors
- Air conditioning
- (Aircraft with CAT-IIIa configuration) Head-up *guidance* system.

The following aircraft systems are interfacing the EICAS for *alerting* purposes only:

- Air turbine starter
- Master announcement
- *Aural* warning
- Central Maintenance Computer (CMC)
- Fire protection
- (E)GPWS/*windshear*
- Rudder
- Smoke
- *Stall* protection system
- Thrust reverser

The integrated architecture of the system uses the EICAS display and MFD's for indications.

The Radio Management Units, two separate displays which are normally used for Radio Management purpose, can be used to display EICAS information (engine parameters and critical messages) as a backup in case of failure of an Integrated Computer.

EICAS Primary Format

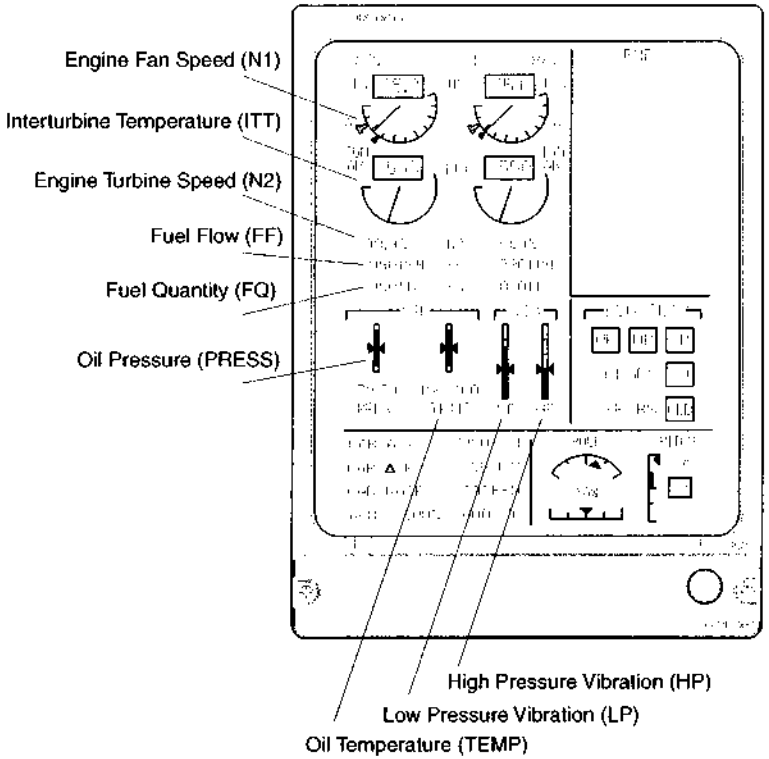
The EICAS primary format is always shown at the DU-3, and is divided in three parts:

- Engine indications
- System indications
- Crew *alerting*

Engine Indication

The most important parameters of the engines are presented on the EICAS primary format on DU 3 as shown on Figure 28.

Figure 28: Engine Indications on EICAS Display



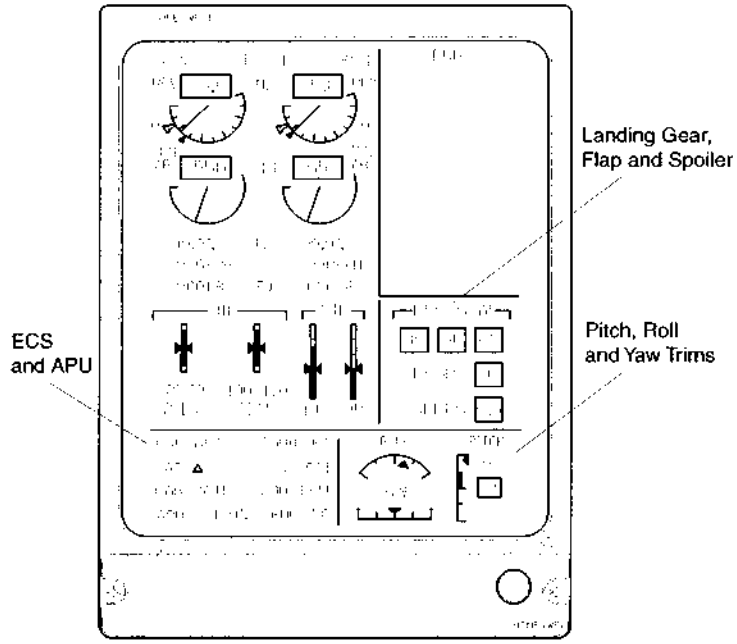
For any exceeded parameter, the indication will become red and flash for 10 seconds and then stay ON.

The indication *bug* follows the same color as found along the analog band

Indication of other Aircraft Systems

Figure 29 highlights the Indication of the other aircraft systems that are continuously shown on the EICAS primary Format.

Figure 29: Indication of other Aircraft Systems on EICAS Display



Crew *Alerting*

The EICAS continuously monitors the status of various aircraft and avionics system, and generates messages for the flight crew members.

Each IC-600 receives warning, caution, *advisory*, and status signals from systems. The IC that controls the EICAS display, which is selected through the reversionary panel, will have the messages displayed.

The CAS area location is at the upper right field of the EICAS display. This field has the capability of 16 lines of 18 characters per line. The last line shows the status line.

More than 150 messages will be shown on the CAS according to the message type. The message types are warning, caution, and *advisory*. Warning (red color) messages are displayed on top, followed by caution (*amber* color) messages, and *advisory* (*cyan* color) messages. The newest message will be displayed on top of its respective *queue*. When a new message occurs, the message will flash.

Message *recognition* is done as follows:

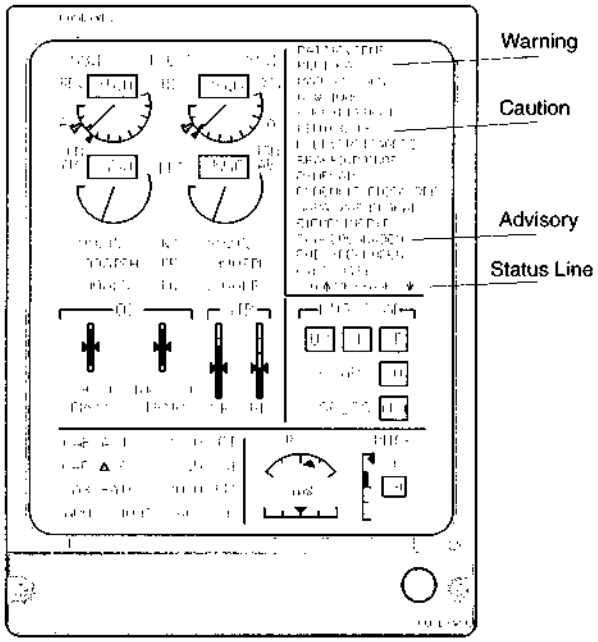
- For warning message: by pushing the master warning button, located on the glareshield panel.
- For caution message: by pushing the master caution button, located on the glareshield panel.
- For *advisory* message: automatically, after five seconds.

The "END" message will always be the last message on the status line.

If there are 15 messages, the 16th line (status line) will show text "XX- messages XX". This text refers to the number of messages scrolled off the top and bottom of the display, respectively.

If there are no messages, then the status line will stay blank. The "END" message will not be included in the count of messages scrolled off the bottom of the display.

Figure 30: Example of Crew *Alerting* Indications on EICAS Display



Components of the EICAS:

The EICAS includes the following components:

- Two DAU's (DAU # 1/2)
- One EICAS display (DU # 3)
- Two MFD's (DU # 2/4)
- Two integrated computers (IC-600)
- Two Radio Management Units (RMU's)
- Two reversionary panels
- One EICAS reversion panel.

Data Acquisition Unit (DAU)

The DAU is the central data collection point for the EICAS. The DAU receives *discrete* and analog inputs. These signals are changed for the digital ones by an analog/digital converter and sent to the integrated computer (IC-600).

The DAU is a dual (A and B) channel unit, the channels being *redundant*, except for the excitation source for signals that do not require it.

EICAS Display

The DU-3 is a full color CRT 8 x 7 inch electronic display, which is identical to the PFD's and MFD's. For these DU's only the *bezels* are completely different either in the format or in the operation.

Multi-function Display - MFD

The MFD (DU 2/4) is similar to DU-3, but, through its menu, you can get access to system pages, as follows:

- Takeoff page
- ECS page
- Fuel page
- Hydraulic/brake page
- Electrical page

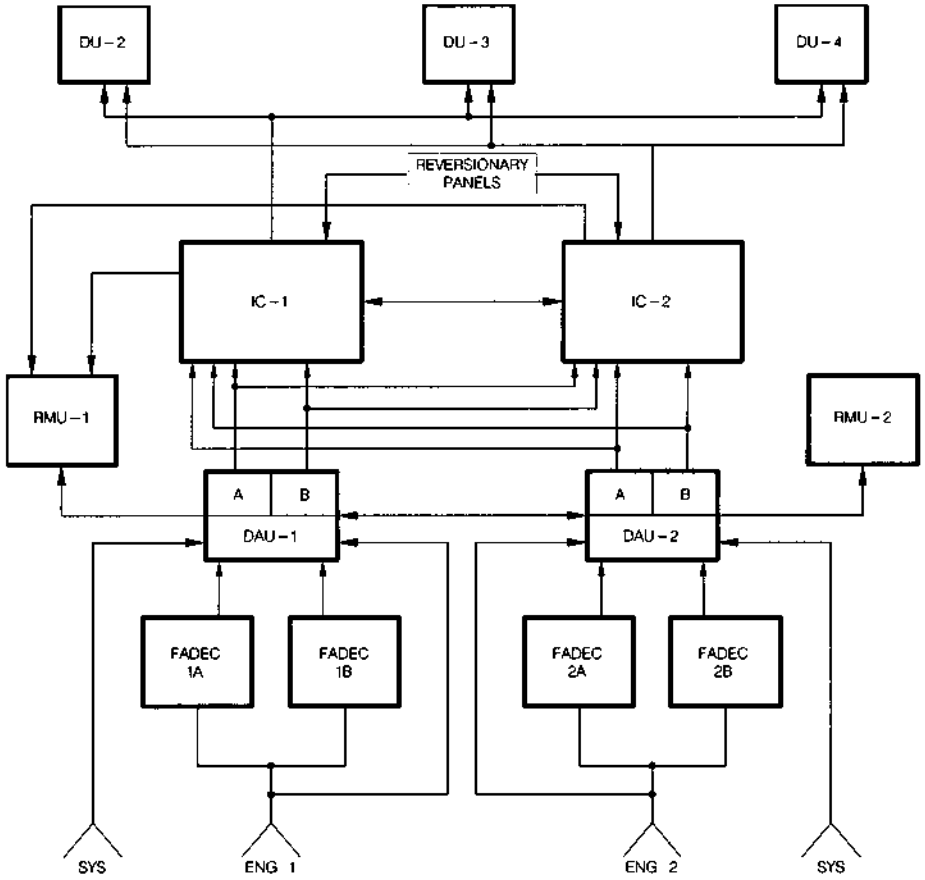
Integrated Computer

The integrated computer (IC-600) is the unit which receives digitized signals (ARINC 429) from DAU's for generating images to DU's.

In the IC-600 there is a symbol generator which controls the display units in normal condition, as follows:

- IC # 1, controls DU's 1, 2 and 3.
- IC # 2, controls DU's 4 and 5.

Figure 31: Architecture of the EICAS System



Radio Management Unit - RMU

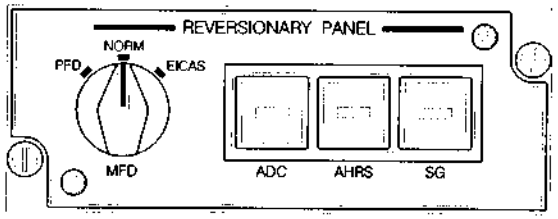
Both RMU's *provide* backup EICAS information through the engine backup page, for normal operation.

Only the RMU # 1 supplies the backup EICAS information in emergency operation.

Reversionary Panel

A reversionary panel is *provided* for each pilot to *permit* Symbol Generator (SG) reversion. It also *permits* Reconfiguration of the MFD either for EICAS display or Primary Flight Display (PFD).

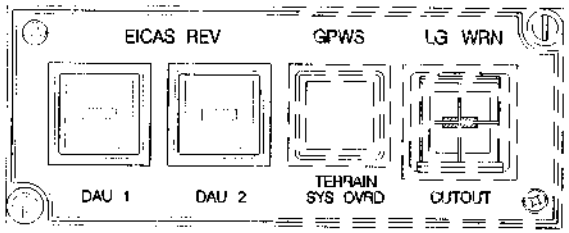
Figure 32:



EICAS Reversion Panel

Through the EICAS reversion panel, you can deenergize the failed channel (only channel A) of DAU 1/2.

Figure 33:



Aural Warning

As on the Airbus A320 Warnings, and Caution messages *triggered* by a system failure will be shown on the display units (EICAS displays for Embraer).

Some Warnings, depending on the seriousness of the failure, will also *trigger* an *aural* warning. On the Airbus A320 the *aural* warnings are part of the ECAM system.

On the Embraer 145, the *aural* warnings are discussed in a separate sub chapter in the AMM called "*Aural* Warning".

This is just a different philosophy of defining the different parts of the systems.

The *relation* between Messages on the screens and *aural* warnings is very similar on all modern aircraft.

As the *aural* warnings are also part of the Electronic Instrument System, a short description of the system will be given on the following pages.

System Description

The *aural* warning system receives signals from the following aircraft systems:

- TCAS (Ref. 34-43)
- *Windshear detection* system
- GPWS/EGPWS
- IC-600
- Fire *detection* system
- *Stall* protection system
- Trims
- Flaps
- Brakes
- Spoilers
- Radio altimeter
- Autopilot

The system generates voice messages, and warning and caution tones *associated* with a failure.

The tones have different forms.

Aural Warning Levels

The *aural* warnings are divided into four *levels*, as follows:

- **Emergency**
  Associated with situations that can be dangerous. The system can generate a voice. If an emergency fault occurs, with voice message, the *Aural* Warning Unit (AWU) generates a specific sound, three chimes, plus the voice message. If an emergency fault occurs, without voice message *associated*, the AWU generates only master warning tone. In any condition, the *aural* warning is repeated every second until cancellation through the master warning light or until the failure is removed.
- **Abnormal**
  Associated with malfunctions or failures. The AWU generates a master caution tone, every five seconds, until cancellation through the master caution light or the failure is removed.
- *Advisory*
  Associated with the least important failures such as *loss* of redundancy or degradation of a system.
- Information
  The information *alerts* are related to information situations.^

The "*AURAL* LEVEL" table shows the *aural* message priorities and levels.

Table 4: *Aural* **Level**

Level	*Associated* Condition/EICAS Messages	Priority	Tone	Voice Message	Cancel
Emergency	*Stall* Condition.	1	Clacker	None	No
	Windshear condition	2	None	*Windshear*	No
	Ground Proximity Condition	3	None	Several Ground Proximity Warning Messages Possible	No
	Traffic Proximity Condition	4	None	Several TCAS Warning Messages possible	No
	Fire in Engine or APU / ENG 1 (2) FIRE, APU FIRE	5	Bell	None	Yes
	Airspeed above VMO	6	three chimes	High Speed	No
	Landing Gear not locked down for landing	7	three chimes	Landing Gear	No
	Cabin *Altitude* above 10,000 ft.	8	three chimes	Cabin	Yes
	Associated with takeoff configuration warning	9	three chimes	Takeoff plus one of the conditions: • Flaps • Brakes • Trim • Spoiler	No
	Associated with emergency failures	10	Master Warning	None	No
Abnormal	*Associated* with *glide slope deviation*	None	None	*Glide Slope*	Yes
	Traffic proximity condition	None	None	Traffic	Yes
Abnormal cont'd	*Associated* with abnormal failures	None	Master Caution	None	Yes

Table 4: *Aural* **Level**

Level	*Associated* Condition/EICAS Messages	Priority	Tone	Voice Message	Cancel
Advisory	Autopilot disengagement during approach	None	None	Autopilot	Yes
	Associated with decision height crossing	None	None	Minimum	No
	Power-up test detected a failure in one channel of AWU	None	None	*Aural* Unit One Channel	No
Information	*Associated* with SELCAL callings	None	None	Selcal	No
	Airplane is crossing or has reached a pre-selected *altitude*	None	2,900 Hz three times	None	No
	Both channels of AWU are operating normally on power-up test	None	None	*Aural* Unit O.K.	No
	Takeoff configuration test successful	None	None	Takeoff O.K.	No
	Power supply 1 or 2 failure	None	None	*Aural* Unit One Power Interrupt	No

If multiple *alerts occur*, the highest *level* (emergency) *alert* operates first. After cancelling the emergency *aural alerts*, the abnormal *alerts* operate. If all emergency and abnormal *alerts* are cancelled, the *advisory alerts* operate. Finally, if there are no other *alerts*, then the information *alerts* operate.

But, any *alert* that is in progress will be completed before another *alert* signal starts to operate (even if the priority is higher).

A special condition occurs in the emergency fault *level* without voice message. "three chimes" is *inhibited* when any other *alert* occurs at the same time. Only after cancelling an *alert* generated by the AWU system, "three chimes" will be generated.

If a *stall* or fire fault occurs at the same time as any emergency fault with voice message *associated*, the two operate *simultaneously*. The amplitude of the *discrete* sound (clacker or bell) is reduced, but is sufficiently distinct to *permit* the crew to understand the voice message and the *discrete* sound easily.

A special condition occurs for *windshear*, TCAS, or GPWS/EGPWS *alert* condition. No other voice message is generated with them to avoid misunderstanding. Only *stall* condition has priority over *windshear*, GPWS/EGPWS, and TCAS *alerts*.

Master Warning and Master Caution

There are two switches to cancel emergency or abnormal failures. They are the master warning light and the master caution light. The *associated* light flashes when an emergency or caution failure occurs. The pilot or copilot can cancel all the *alerts* through the master warning/caution lights, except:

- *Stall*
- Landing configuration alarm generated because of the flaps
- GPWS/EGPWS
- *Windshear*
- TCAS R/A, Overspeed, *Advisory* and information *alerts*

Interfaces to the Aural Warning System

The *aural* warning system makes interface with:

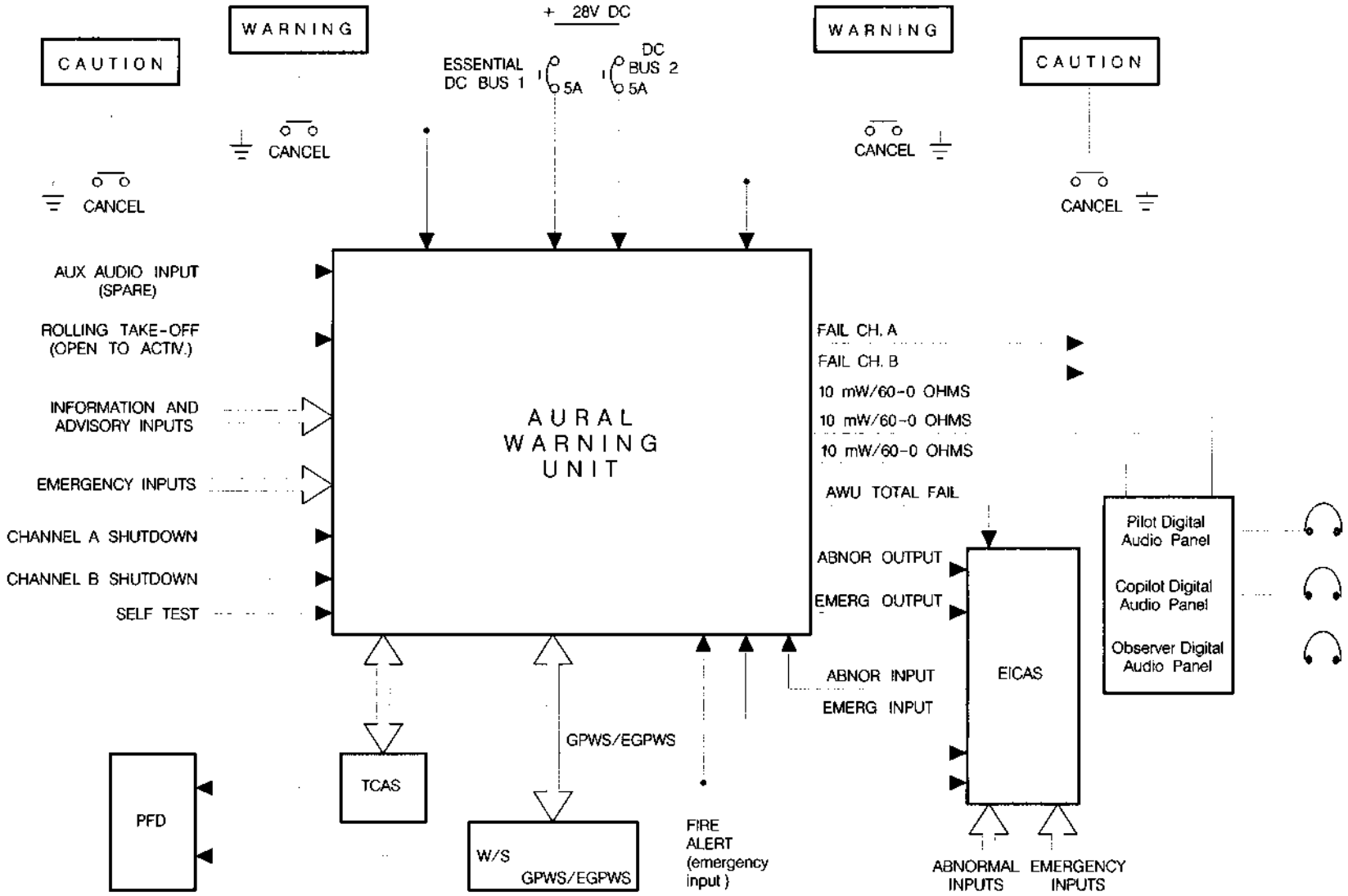
- Master warning and master caution light inputs
  They are used to cancel a warning or a caution fault. A ground signal comes through the switches.
- TCAS auxiliary audio input
  There is an auxiliary audio input for the audio generated in the TCAS computer.
- (E)GPWS/*windshear* auxiliary audio input There is an auxiliary audio input for the audio generated in the GPWS/EGPWS and W/S computer.

NOTE: The AWU uses these interfaces to give priority of *alerts*.

- Rolling take-off configuration input
  This input is used for a take-off configuration simulation during aircraft taxi. There is a switch in the control pedestal to simulate a condition of power-levers in the take-off configuration. This function *permits* the crew to do a take-off configuration test during aircraft taxi.
- Emergency, *advisory* and information inputs
  These inputs are connected to some systems, namely:
  - SELCAL
  - *Altitude alert*
  - Autopilot
- Pilot/copilot and observer digital audio panels outputs
  The *aural* warning system interfaces the 3 audio panels to have head phone and speaker outputs. The speaker output, for warning purposes, on the pilot and copilot audio panels can never be switched off from crew members' action.
- Abnormal and emergency inputs The two inputs come from EICAS when an emergency or abnormal fault occurs *associated* only to the CAS messages on the EICAS display. These inputs will cause the AWU to generate master warning tone (three chimes) and master caution tone.
- Abnormal and emergency outputs
  The two outputs go to the EICAS when the AWU is ready to generate the audio signal. Even when the master warning light of master caution light is pushed, the AWU sends a ground signal to cancel the light flashing.
- AWU total fail output
  This output informs the EICAS if the AWU fails totally.

Figure 34 shows the interfaces with the different aircraft systems

Figure 34: *Aural* **Warning Block Diagram**



5.2 Numbering Systems

Introduction

General

In the aeronautical world switching circuits of one kind or another are to be found in almost every piece of electronic equipment ranging from ground radar systems, aircraft navigation and flight systems and communications equipment. The switching systems may be relatively simple or quite complex, *incorporating* many *interrelated* circuits in the form of 'gates', 'logic circuits' and digital computer units.

It is necessary to understand the numbering system on which these circuits operate and to be able to interpret indications which may display those numbering system 'symbols' (*digits* and letters). We shall first look at the various numbering systems, which are used:

- Decimal
- Binary
- Octal
- Hexadecimal

Each numbering system will be examined for its structure and conversion to other systems.

Definition of Base or Radix

The number of symbols used in a numbering system is determined by the *base* or *radix* of the system, e.g. 10 *digits* (0,1,2,3,4,5,6,7,8,9) are used in the decimal system and is called a *base* (or *radix*) 10 system.

A binary system only uses two symbols (the *digits* 0 and 1) and is therefore called a *base* 2 system.

When dealing with computer systems it is often *convenient* to work with octal which is a *base* 8 system (using the *digits* 0,1,2,3,4,5,6,7) or hexadecimal which is a *base* 16 system (using the *digits* 0,1,2,3,4,5,6,7,8,9 and the letters A,B,C,D,E,F).

Unless the *base* of the system being used is clear, a *subscript* of the *base* should be used, e.g.:

Decimal	123_{10}
Binary	1111011_2
Octal	173_8
Hexadecimal	$B7_{16}$

Note:

An expert in numbering calculations (graduated to that title after a successful study of this JAR-66 module 5.2) should be able to check the conversions from the decimal number 123 to the *equivalent* binary-, octal- and hexadecimal values, as shown in the example before.

Please, space is given here below and in the blank spaces in the right columns of the following pages!

Decimal System

Notes:

General

In this system the *base* is 10. *Consider* the number 7245_{10}.

This is *stated* as seven thousand, two hundred and forty five.

Written in its full form it becomes:

$(7 \times 10^3) + (2 \times 10^2) + (4 \times 10^1) + (5 \times 10^0)$.

The 10^3, 10^2, 10^1 and 10^0 are 'understood' but each of them is *prefixed* by the *appropriate* coefficient, in this case 7, 2, 4 and 5. It can therefore be seen that each *power of* 10 can have a coefficient ranging from 0 to 9, again ten *digits*.

Disadvantages of the Decimal System

In a digital computer, calculations are carried out in a *central processor unit (CPU)* which may *consist* of many transistors, diodes etc. If a single transistor is being used to count on the decimal scale of *base* 10, then 10 *discrete* levels of collector current would be required, e.g.:

0 mA might represent an input of 0, 1 mA an input of 1, 2 mA an input of 2, etc.

Such a system would be extremely difficult to operate and would require very careful design with stabilised power supplies, negative feed back networks, close tolerance components etc. One *requirement* of a digital computer is that it is *accurate* and therefore it must be able to *distinguish* between numbers even if they are close to each other. In addition it is important that the number does not *drift* with changes in power supply voltage or temperature. It would be risky to expect a transistorised circuit to *distinguish infallibly* between 10 different *magnitudes* of current required by a decimal system.

Where *accuracy* and speed are important there are only 2 possible *states* in an electronic device on which *adequate reliance* can be placed: These are the on *state* and the off *state*. These *states* are the basis of the system, which is used almost universally today throughout digital *processing* and is called the binary *base* 2 system. By using two *widely* spaced current values *corresponding* to the output of a transistor at *saturation* (on) and *cut-off* (off), the chance of error in representing a given number is minimised.

The on/off *state* of a transistorised circuit represents the binary numbers 0 and 1 and since they are well defined and can be *distinguished* readily from each other, even relatively large *drifts* in power supply voltage or operating temperature will not affect them.

Binary Numbering System

General

A decimal number, when written down, does not have the *base* numbers 10^0, 10^1, 10^2 etc. included but are 'understood' to be there. Furthermore the coefficient of each *power of* ten ranged from 0 to 9.

In a binary number the *base* numbers (*power of* two) 2^0, 2^1, 2^2, etc. are not shown - similar as within decimal numbers - and the coefficient of each *base* number can only be 0 or 1.

To explain this the binary number 1 1 0 1 0 1 is illustrated in the top row of the table below, underneath of each binary *digit* is written its *base* number (*power of* two) and in the last row of the table appears the decimal *weight* of each column.

1	1	0	1	0	1	Binary Number
2^5	2^4	2^3	2^2	2^1	2^0	*Base* Number
32	16	8	4	2	1	Decimal *Weight*

Conversion of a Binary Number to its Decimal Equivalent

To *convert* the above binary number 1 1 0 1 0 1 to decimal form, the decimal *weight* of each of the *base* numbers having a coefficient of 1 (binary *digit* = 1) are added together giving:

32 + 16 + 0 + 4 + 0 + 1 = 53

Note that 2^0 equals 1 which is true for any number having an index of 0.

This is one way in which any binary number can be *converted* into its *equivalent* decimal number. This system is *acceptable* for small binary numbers, i.e., numbers with a small number of 0's and 1's but becomes *tedious* when having to deal with large binary numbers.

Under these *circumstances* it is better to use the following *successive* multiplication technique to *convert* a binary number to a decimal.

This *conversion* takes place as follows:

- Multiply the most *significant digit* (the left-most *digit*) by 2 (*base* or radix).
- Add the following binary *digit* of the binary number to this first product then multiply by 2 again.
- Add the next *digit* to this product then multiply by 2 again. This procedure continues until the least *significant digit* is added to the products generated, at this point the *conversion* is completed.

	MSB					*LSB*
Binary Number	1	1	0	1	0	1
Conversion sums from multiplying by 2 and adding next binary *digit*.	2	3	6	13	26	53

Note:

A binary number can have 0's placed in front of the first 1 (left-most *digit*) of the number without altering its decimal value.

It should also be noted that the first (left-most) *digit* of a binary number is called '*MSB*' (**M**ost **S**ignificant **B**it) and the last (right-most) *digit* is called '*LSB*' (Least **S**ignificant **B**it). The word **'bit'** comes from the first two letters of **bi**nary and last letter of *digit*. The '*MSB*' and '*LSB*' may also be called 'MSD' and 'LSD' (..D = *digit*).

Apropos, as shown in the table below with the binary *equivalent* of 163_{10}, four bits make a *Nibble* and two *Nibbles*, or eight bits, make a **Byte**. These groupings are useful when *converting* to hexadecimal notation, which is explained later.

163_{10}	=	128	+0	+32	+0	+0	+0	+2	+1
		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
10100011_2	=	1	0	1	0	0	0	1	1
		MSB							*LSB*

¦<---------- *Nibble* ---------->¦ ¦<---------- *Nibble* ---------->¦

¦<---------- Byte ---------->¦

Conversion of a Decimal Number to its Binary Equivalent

Since the binary number system is the only one which a digital computer *recognises*, it is necessary to *convert* decimal numbers into their binary form.

To *convert* from decimal to binary, *successive* division by 2 may be *employed* and the *remainder* of any division (which must be either 0 or 1) is then recorded *successively* in a separate column. The division by 2 is continued until the number being divided yields an answer of 0 with a *remainder* of 1.

The binary number *equivalent* of the original decimal number is then taken as the sequence of *remainders* reading from the bottom upwards. Clearly, since the last divis.

The following is an example of how this conversion is carried out:

Write the decimal number 217 in binary form.

2	217	*Remainders*	
2	108	1	*LSB*
2	54	0	
2	27	0	
2	13	1	
2	6	1	
2	3	0	
2	1	1	
	0	1	*MSB*

Read up to obtain binary *equivalent*:
1 1 0 1 1 0 0 1

i.e. $217_{10} = 11011001_2$

Notes:

Octal Numbering System

General

As the name implies, the octal system is a *base* 8 system, using the symbols 0, 1, 2, 3, 4, 5, 6 and 7. The *weight* factors are now powers of eight, such as 8^0, 8^1, 8^2, etc. and the coefficient of each *base* number can be in the range of 0 to 7.

To explain this the octal number 236 is illustrated in the top row of the table below, underneath of each octal *digit* is written its *base* number (*power of* eight) and in the last row of the table appears the decimal *weight* of each column.

2	3	6	Octal Number
8^2	8^1	8^0	*Base* Number
64	8	1	Decimal *Weight*

Conversion of an Octal Number to its Decimal Equivalent

To *convert* the above octal number 236 to decimal form, the decimal *weights* of each of the *base* numbers having a coefficient of 1 to 7 have to be multiplied by the respective octal *digit* of the octal number and are added together giving:

2 x 64 + 3 x 8 + 6 x 1 = 158

Another method to *convert* an octal number to its decimal *equivalent* is the *successive* multiplication technique.

This conversion takes place as follows:

- Multiply the most significant *digit* (the left-most *digit*) by 8 (*base* or radix).
- Add the following octal *digit* of the octal number to this first product then multiply by 8 again.
- Add the next *digit* to this product then multiply by 8 again. This procedure continues until the least significant *digit* is added to the products generated, at this point the conversion is completed.

The *successive* multiplication technique in *converting* an octal number to its decimal *equivalent* is shown in the table below with the well-known octal number 236.

Octal Number	2	3	6
Conversion sums from multiplying by 8 and adding next octal *digit*.	16	19	158

Conversion of a Decimal Number to its Octal Equivalent

To *convert* decimal to octal, divide the decimal number by 8 *successively* and again record the *remainders*. As in the conversion from decimal to binary the answer is then read upwards!

Example:

Convert the decimal number 796 to its octal *equivalent*:

8	796	*Remainders*	
8	99	4	LSD
8	12	3	
8	1	4	
	0	1	MSD

Read up to obtain octal *equivalent*: 1434

i.e. $796_{10} = 1434_8$

Conversion of an Octal Number to its Binary Equivalent

This octal number 1434 will now be *converted* to its binary *equivalent* which is *accomplished* by taking each octal *digit* and *converting* it to a three-bit group as follows:

1	4	3	4	Octal Number
001	100	011	100	3-Bit Groups

i.e. $1434_8 = 001\ 100\ 011\ 100_2 = 1100011100_2$

Taking these results altogether gives: $796_{10} = 1434_8 = 1100011100_2$

Conversion of a Binary Number to its Octal Equivalent

In order to *convert* a binary number to its octal *equivalent* the binary number *digits* are split into groups of three bits, starting from the least significant bit (*LSB* = rightmost bit). *Noughts* are then added in front of the most significant bit (*MSB* = leftmost bit), if required to complete the group of 3.

For example:

Convert the binary number 10110011 to its octal *equivalent*.

Starting from the *LSB* and putting the *digits* in groups of 3 gives: 10 110 011.

A *nought* must be placed in front of the *MSB* to complete the group of 3 before *converting* each group to its octal value, i.e.

010	110	011	3-Bit Groups
2	6	3	Octal Number

Therefore: $10110011_2 = 263_8$

Notes:

Hexadecimal Numbering System

General

This system is used mainly on those systems handling large binary numbers and, as with the octal system, it is used as a shorthand method of writing binary numbers. This system uses a *base* of 16.

However, since only ten different *digit* symbols exist (0 to 9), six other symbols have to be used for the complete range of coefficients of each *power of* 16. These are the letters A to F inclusive of the alphabet. The complete set of hexadecimal symbols of each *base* number (16^0, 16^1, 16^2, etc.) is now:

0	...	9	A	B	C	D	E	F	Hexadecimal Symbols
0	...	9	10	11	12	13	14	15	Decimal *Weight*

To explain this the hexadecimal number 1A3 is illustrated in the top row of the table below, underneath of each hexadecimal symbol is written its *base* number (*power of* 16) and in the last row of the table appears the decimal *weight* of each column.

1	A	3	Hexadecimal Number
16^2	16^1	16^0	*Base* Number
256	16	1	Decimal *Weight*

Conversion of a Hexadecimal Number to its Decimal Equivalent

1 x 256 + 10 x 16 + 3 x 1 = 419

The table above illustrates the conversion of the hexadecimal number 1A3 to decimal form: The decimal *weights* of each of the *base* numbers having a coefficient of 1 to 9 or A to F have to be multiplied by the respective hexadecimal symbol of the hexadecimal number and are added together.

Another method to *convert* a hexadecimal number to its decimal *equivalent* is the *successive* multiplication technique.

This conversion takes place as follows:

- Multiply the most significant symbol (the left-most symbol) by 16.
- Add the following hexadecimal symbol of the hexadecimal number to this first product then multiply by 16 again.
- Add the next symbol to this product then multiply by 16 again. This procedure continues until the least significant symbol of the hexadecimal number is added to the products generated, at this point the conversion is completed.

The *successive* multiplication technique in *converting* a hexadecimal number to its decimal *equivalent* is shown in the table below with the well-known hexadecimal number 1A3.

Hexadecimal Number	1	A	3
Conversion sums from multiplying by 16 and adding next hexadecimal symbol.	16	26	419

We *recognize* that similar methods - as within the octal system - can be used to *convert* hexadecimal to decimal and we will see, also vice versa from decimal to hexadecimal or the conversions between hexadecimal and binary are performed in a similar way to the octal system.

Conversion of a Decimal Number to its Hexadecimal Equivalent

The decimal number has to be divided *successively* by 16 and the *remainders* are again read upwards to give the hexadecimal number.

For example: *Convert* the decimal number 762 to its hexadecimal *equivalent*:-

16	762	*Remainders*
16	47	10_{10} = A
16	2	15_{10} = F
	0	2_{10} = 2

Read up to obtain hexadecimal *equivalent*: 2FA

i.e. 762_{10} = $2FA_{16}$

Conversion of a Hexadecimal Number to its Binary Equivalent

This conversion is carried out in the same way as for octal to binary except now each hexadecimal symbol is given its *equivalent* binary number as a group of 4.

So the hexadecimal number just found becomes:

2	F	A	Hexadecimal Number
0010	1111	1010	4-Bit Groups

Note: F and A are 15_{10} and 10_{10} respectively.

The complete conversion therefore is:

$762_{10} = 2FA_{16} = 0010\ 1111\ 1010_2 = 1011111010_2$

Conversion of a Binary Number to its Hexadecimal Equivalent

To do this conversion the same procedure as for octal is carried out but the binary *digits* are grouped in fours starting from *LSB* and adding *noughts* as required in front of *MSB* to complete the group of 4.

Example:

Convert the binary number 11011100101 to its hexadecimal *equivalent* number.

Group the *digits* in fours starting from the *LSB*. A *nought* has to be added in front of the *MSB* to complete the grouping, thus: 0110 1110 0101.

Now put the *appropriate* symbol (*digit* or letter) for each group. This gives as the answer:

0110	1110	0101	4-Bit Groups
6	E	5	Hexadecimal Number

Therefore: $11011100101_2 = 6E5_{16}$

Relationship between Binary, Octal and Hexadecimal System

Consider the conversion of the decimal number 1254 into the 3 numbering systems, they are:

- Binary10011100110_2,
- Octal 2346_8,
- Hexadecimal$4E6_{16}$.

It can be seen that for ease of use the hexadecimal is best, the octal next and the last are binary.

The table shown below illustrates the *relationship* between decimal, binary, octal and hexadecimal numbers.

Decimal		**Binary**					**Octal**		**Hexadecimal**		**Numbering System**
10^1	10^0	2^4	2^3	2^2	2^1	2^0	8^1	8^0	16^1	16^0	**Base Number**
10	1	16	8	4	2	1	8	1	16	1	**Decimal** *Weight*
	0	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	1	0	1	0	1	
	2	0	0	0	1	0	0	2	0	2	
	3	0	0	0	1	1	0	3	0	3	
	4	0	0	1	0	0	0	4	0	4	
	5	0	0	1	0	1	0	5	0	5	
	6	0	0	1	1	0	0	6	0	6	
	7	0	0	1	1	1	0	7	0	7	
	8	0	1	0	0	0	1	0	0	8	
	9	0	1	0	0	1	1	1	0	9	
1	0	0	1	0	1	0	1	2	0	A	
1	1	0	1	0	1	1	1	3	0	B	
1	2	0	1	1	0	0	1	4	0	C	
1	3	0	1	1	0	1	1	5	0	D	
1	4	0	1	1	1	0	1	6	0	E	
1	5	0	1	1	1	1	1	7	0	F	
1	6	1	0	0	0	0	2	0	1	0	

Note:
To obtain the octal number *digits*, the *equivalent* binary number has to be divided in 3-bit groups and to obtain the hexadecimal number symbols (*digits* and/or letters) the *equivalent* binary number has to be divided in 4-bit groups, always starting from the *LSB* (2^0).

Binary Coded Decimal (BCD)

A binary number system representation is the most *appropriate* form for internal computations since there is a direct mathematical *relationship* for every bit in the number. To interface with a user - who usually wants to see I/O (Input/Output) in terms of decimal numbers - other codes are more useful. The Binary Coded Decimal (BCD) system is the simplest and most *widely* used form for inputs and outputs of user-oriented digital systems.

In the Binary Coded Decimal (BCD) system, each decimal *digit* is expressed as a *corresponding* 4-bit binary number. In other words, the decimal *digits* 0 to 9 are encoded as the bit *strings* 0000 to 1001. To make the number easier to read, a space is left between each 4-bit group. For example, the decimal number 163 is *equivalent* to the BCD number 0001 0110 0011, as shown in the following table:

BCD	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**1**
	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
	2^0				$2^2 + 2^1$				$2^1 + 2^0$			
Decimal	**1**				**6**				**3**			

A *generic* code could use any n-bit *string* to represent a piece of information. BCD uses 4 bits because that is the minimum needed to represent a 9. All four bits are always written, even a decimal 0 is written as 0000 in BCD.

The important difference between BCD and the *previous* number systems is that, starting with decimal 10, BCD loses the standard mathematical *relationship* of a *weighted* sum. *Instead* of using the 4-bit code *strings* 1010 to 1111 for decimal 10 to 15, BCD uses 0001 0000 to 0001 0101. There are other n-bit decimal codes in use (see next column).

BCD is the simplest way to *convert* between decimal and a binary code. Thus it is the ideal form for I/O interfacing. The binary number system, since it maintains the mathematical *relationship* between bits, is the ideal form for the computer's internal computations.

Gray Code and 2 out of 5 Code

The gray code is often used in analog to digital *converters*, which develop these codes by *converting* an analogically changing parameter value to *digitized* binary pulses (e.g. a sensing element turning an *encoding* disc by *predefined* degrees of rotation). The difference compared with the natural binary- or BCD code is, that only one bit changes the *state* (going from "1" to "0" or vice-versa) when the analogically changing parameter has increased or decreased *beyond* a *predefined* measurement. The advantage of this coding is an easy error *detection*: When two bits change after a parameter 'step' above or below the *predefined* value, something is wrong with the data.

Similar advantages are encountered with the 2 out of 5 code, used on frequency selection devices. Here, always two of five bits (wires) are "1" (open or grounded, dependent on the *assigned* logic) and three are "0". An *increment* or *decrement* of a selected *digit* results in a bit change from "1" to "0" and another from "0" to "1".

Decimal	Binary	Gray	2 out of 5
0	0000	0000	10010
1	0001	0001	00011
2	0010	0011	00101
3	0011	0010	00110
4	0100	0110	01010
5	0101	0111	01100
6	0110	0101	10100
7	0111	0100	11000
8	1000	1100	01001
9	1001	1101	10001

ASCII-Code

To get information into and out of a computer, we need more than just numeric representations; we also have to take care of all the letters and symbols used in day-to-day *processing*. Information such as names, addresses, and item descriptions must be input and output in a readable format. But remember that a digital system can deal only with 1's and 0's. Therefore, we need a special code to represent all alphanumeric data (letters, symbols, and numbers).

Most industry has settled on an input/output (I/O) code called the American Standard Code for Information Interchange (ASCII). The ASCII code uses 7 bits to represent all the alphanumeric data used in computer I/O. Seven bits will *yield* 128 different code combinations, as listed in Table 1.

Each time a key is depressed on an ASCII keyboard, that key is *converted* into its ASCII code and *processed* by the computer. Then, before outputting the computer contents to a display terminal or printer, all information is *converted* from ASCII into standard English.

To use the table, place the 4-bit group in the least *significant* positions and the 3-bit group in the most *significant* positions.

Table 1:

Bits	6-4 >	000	001	010	011	100	101	110	111
3-0 >	0000	NUL	DLE	SP	0	@	P	`	p
	0001	SOH	DC1	!	1	A	Q	a	q
	0010	STX	DC2	"	2	B	R	b	r
	0011	ETX	DC3	#	3	C	S	c	s
	0100	EOT	DC4	$	4	D	T	d	t
	0101	ENQ	NAK	%	5	E	U	e	u
	0110	ACK	SYN	&	6	F	V	f	v
	0111	BEL	ETB	'	7	G	W	g	w
	1000	BS	CAN	(	8	H	X	h	x
	1001	HT	EM	)	9	I	Y	i	y
	1010	LF	SUB	*	:	J	Z	j	z
	1011	VT	ESC	+	;	K	[	k	{
	1100	FF	FS	,	<	L	\	l	|
	1101	CR	GS	-	=	M	]	m	}
	1110	SO	RS	.	>	N	^	n	~
	1111	SI	US	/	?	O	—	o	DEL

Note: Bit 6 = *MSB*; Bit 0 = *LSB*; e.g. letter A = 1000001.

Alphabetical list of the ASCII function commands:

ACK	acknowledge	FF	form feed
BEL	bell	FS	file separator
BS	backspace	GS	group separator
CAN	cancel	HT	horizontal tab
CR	carriage return	LF	line feed
DC1	device control 1	NAK	negative acknowledge
DC2	device control 2	NUL	null
DC3	device control 3	RS	record separator
DC4	device control 4	SI	shift in
DEL	delete	SO	shift out
DLE	data link espace	SOH	start of *heading*
ENQ	enquiry	SP	space
EM	end of medium	STX	start of text
EOT	end of transmission	SUB	substitute
ESC	escape	SYN	synchronous idle
ETB	end of transmission block	US	unit separator
ETX	end of text	VT	vertical tab

Summary

In this submodule we have learned that

1. Numerical quantities *occur* naturally in analog form but must be *converted* to digital form to be used by computers or digital circuitry.
2. The binary numbering system is used in digital systems because the 1's and 0's are easily represented by ON or OFF transistors, which output for example 0V for 0 and 5V for 1.
3. Any number system can be *converted* to decimal by multiplying each *digit* by its *weighting* factor.
4. The *weighting* factor of the least *significant digit* in any numbering system is always 1.
5. Binary numbers can be *converted* to octal by forming groups of 3 bits and to hexadecimal by forming groups of 4 bits.
6. The *successive*-division procedure can be used to *convert* from decimal to either binary, octal, or hexadecimal.
7. The binary-coded-decimal system uses groups of 4 bits to drive decimal displays such as those in a calculator.
8. ASCII is used by computers to represent all letters, numbers, and symbols in digital form.

5.3 Data Conversion

Introduction

Most naturally *occurring* physical quantities in our world are **analog** in nature. An analog signal is a continuously variable electrical or physical quantity. Think about a mercury-filled tube thermometer; as the temperature rises, the mercury expands in analog fashion and makes a smooth, continuous motion relative to a scale measured in degrees. The velocity and force with which a musician strikes a piano key are analog in nature. The resulting vibration of the piano *string* is an analog, sinusoidal vibration.

In aviation technology, feedback signals from control surfaces, or pressure signals from static ports or pitot tubes used to get aircraft *altitude* and *airspeed* information, are analog voltage signals. This are just a few examples for all the analog signals we are dealing with on an aircraft or anywhere in the world.

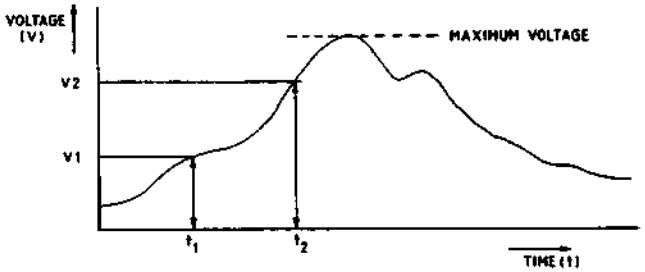
So why do we need to use **digital** representations in a world that is naturally analog? The answer is that if we want an electronic machine to interpret, communicate, and store analog information it is much easier for the machine to handle it if we first *convert* the information to a digital format. A digital value is represented by a combination of ON and OFF voltage *levels* that are written as a *string* of 1's and 0's.

For example, an analog thermometer that registers 26 degrees can be represented in a digital circuit as a series of ON and OFF voltage *levels*. We have learnt in a *previous* section that the number 26 *converted* to digital *levels* is 0001 1010.

Analog signals

Analog signals have any voltage *level* or maximum values. *Level* or amplitude changes are slow or fast but always smooth and continuous.

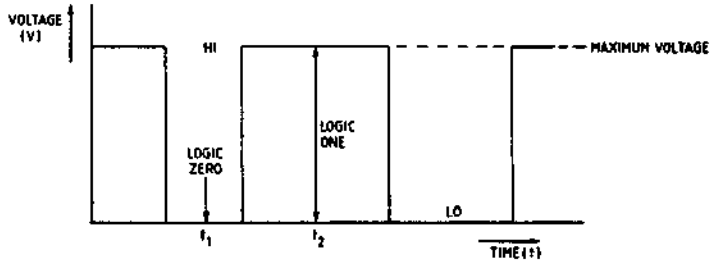
Figure 1: Analog Signals



Digital signals

Digital signals only have two possible *levels*, logic-one (HI) and logic-zero (LO). *Level* changes of digital signals are very fast. They change from high to low and from low to high in a split second.

Figure 2: Digital Signals



A good example of the use a digital representation of an analog quantity is the recording of music. Compact Disks (CD's) and digital audio tapes are becoming usual and are proving to be superior means of recording and playing back music. Musical instruments and the human voice produce analog signals and the human ear naturally responds to analog signals. So, where does the digital format fit in? Although the *process* requires what appears to be extra work, the recording industries *convert* analog signals to a digital format and then store the information on a CD or DAT. The CD or DAT player then *converts* the digital *levels* back to their *corresponding* analog signals before playing them back for the human ear.

While digital computers *process* information faster and more efficiently than analog circuits, they have the disadvantage that they only understand 1's and 0's. Therefore an interface is needed between the digital computers and the analog world. These interfaces are referred to as digital to analog (D/A) *converters* and analog to digital (A/D) *converters*.

D/A *converters* change digital data words from a digital computer to an *equivalent* analog signal as either a voltage or a current source. These analog voltages or currents are then used for a further activation within a system. For example drive a motor or produce sound.

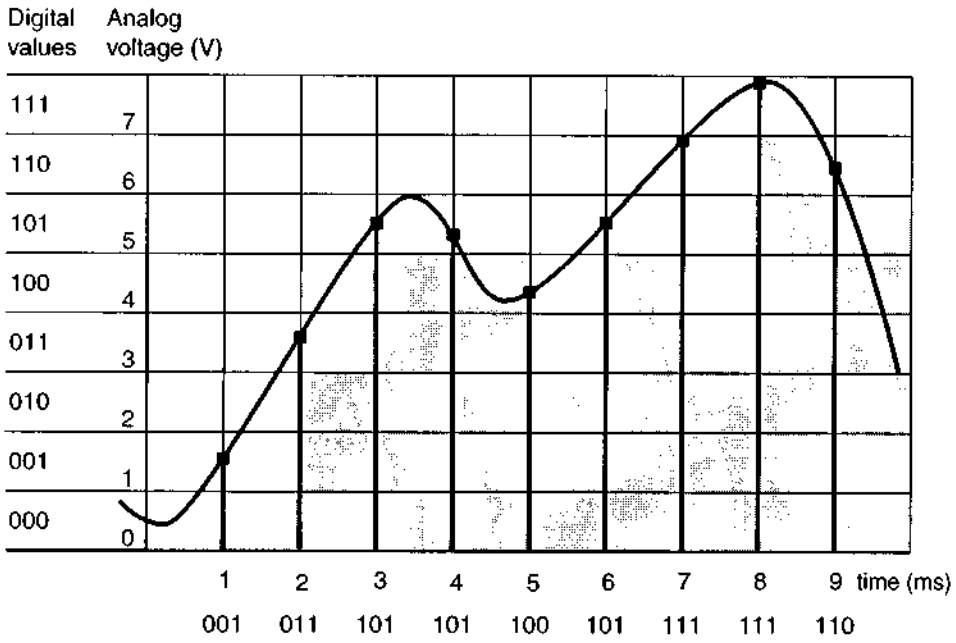
A/D *converter* change the analog signals from sensors to digital data that can be understood by the digital computer. There are many different types of A/D *converters*; which type is used depends on the type of analog input and on what the digital computer needs to know about the analog signal.

Analog and Digital Representation

In an A/D *converter*, a range of input values must *correspond* to a *unique* digital word. The type of code used depends on the system but here only binary coding will be *considered*.

Consider an analog signal that can take on any value between 0 and 8 volts. For any particular voltage, there is a *corresponding* binary code word. For example, using 3-bit words, the voltage analog value between 4 and 5 volts would be represented in binary code by the word 100 that would change to 101 when the analogue value passed through 5 volts. See Figure 3, which illustrates this.

Figure 3: Analog and Digital Representation: Voltage Versus Time



Digital values corresponding to the quantisation levels

The *levels* at which the code changes are known as *quantisation levels*, and the intervals between them as quantisation intervals. In the example given in Figure 3, the *quantisation levels* are 0, 1, 2, 3, 4, 5, 6 and 7 volts, and the quantisation interval is 1 volt.

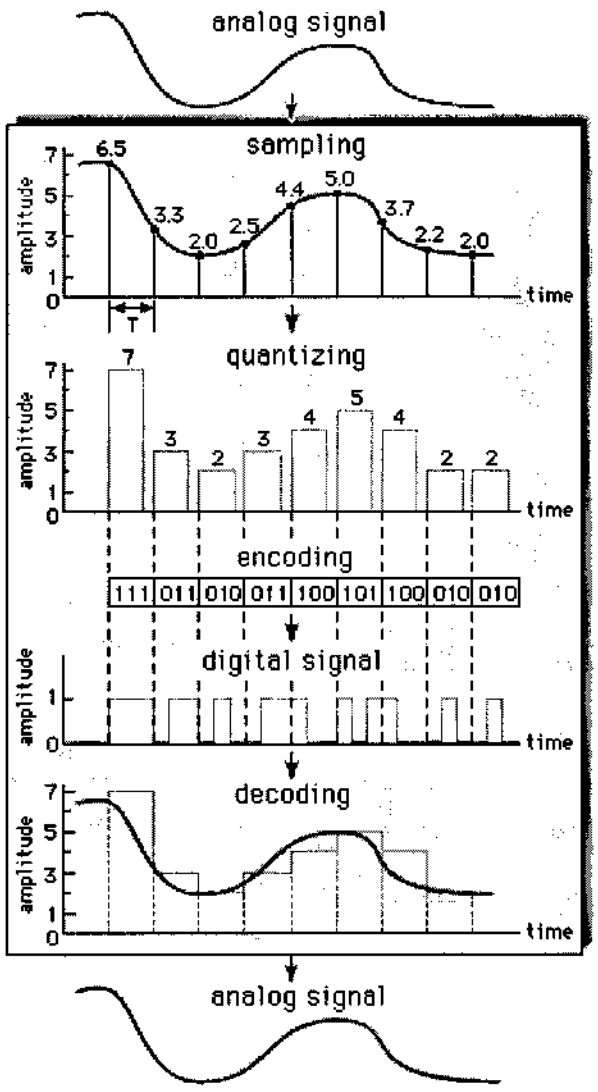
Using a 3-bit word gives $2^3 = 8$ different *quantisation levels*. A 4-bit word would have $2^4 = 16$ *quantisation levels* with 0.5 volt quantisation intervals giving *improved resolution* over the same range of input voltage.

Thus, the more bits available the greater the *resolution* for a given range of analog signal input. It can be seen from the above that an A/D *converter* using an n-bit word would have a *resolution* of one part in 2^n.

In an **D/A** *converter* the conversation is *based* on the same principle, but the *process* is reversed. The input is a digital data word and the output is an analog value that depends on the number of bits set to 1 in the incoming data word.

In the following section different type of D/A and A/D *converters* will be explained. As the D/A *converter* is used as a part of the A/D *converter*, it is necessary to understand the principle of the D/A conversion first.

Figure 4: Example of Digital to Analog and Analog to Digital Conversion



Digital to Analog Converters

Purpose

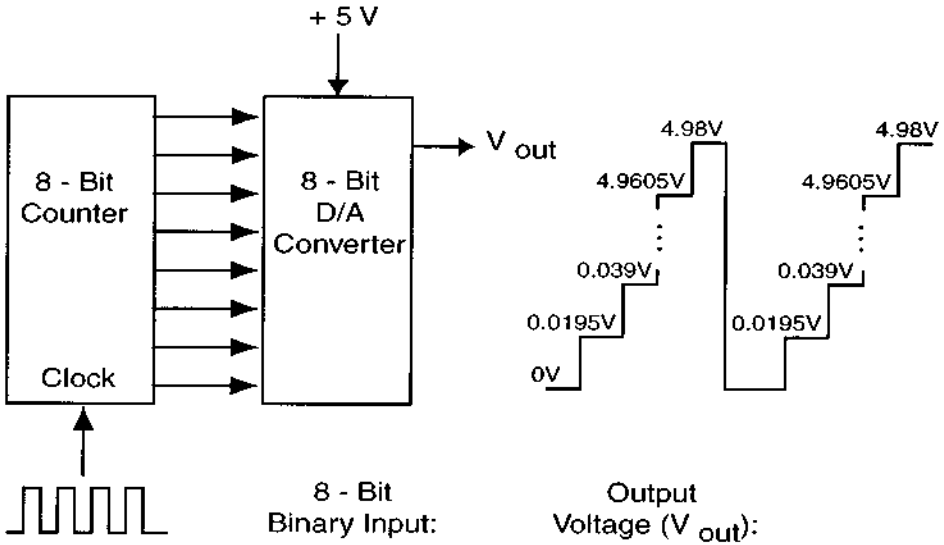
Each D/A circuit has a specific purpose *based* on its use within the system but the general purpose of all D/A *converters* is to *provide* an analog signal output *based* on the digital value represented in the digital computer. This analog signal is either a voltage or current output but the principles involved are essentially the same.

Basic Converter Principle

The basic principle of a D/A *converter* is to divide the analog output into a series of small steps. The number of steps depends on the number of bits used in the data to be *converted*. If the data *consists* of 8 bits then the output is divided into 256 (2^8) steps. The size of each step depends on the range of the *converter*. For a *converter* with a range of 0 - 5 Volts each step would be 0.0195 Volts (5V / 256 steps). If an 8 bit, 5 volt *converter* is driven by a simple counter then the output of the *converter* would be a series of 256 steps of 0.0195 volts.

As the counter progresses from 0 to 255, the *converter* output increases from 0 Volts to 4.98 Volts. It then *drops* to zero when the counter rolls over. Also, note that the maximum output voltage is not 5 volts. This is due to the fact that each digital input bit is *weighted* according to its position within the binary input. The least significant bit (*LSB*) has a *weight* of 5V / 256 = 0.0195 volts, the next most significant bit has a *weight* of 5V / 128 = 0.039 volts, the next has a *weight* of 5V / 64 = 0.078 Volts, and so on, with the most significant bit (*MSB*) having a *weight* of 5V / 2 = 2.5 volts. If you add all the individual bit *weights*, you get a 4.98 volt maximum output when the D/A input is 1111 1111.

Figure 5: Basic Conversion Principle



8 - Bit Binary Input:	Output Voltage (V_{out}):
0000 0000	0.0000V
0000 0001	0.0195V
0000 0010	0.0390V
0000 0011	0.0585V
0000 0100	0.0780V
⋮	⋮
1111 1101	4.9410V
1111 1110	4.9605V
1111 1111	4.9800V

Binary Weighted Ladder

General Description

The binary *weighted Ladder* network is one of the simplest D/A *converters*. Each bit of a binary input controls a solid *state* switch (D_0-D_3) which connects either a reference voltage V_{ref} (5V in Figure 6) or ground to the *corresponding* binary *weighted* resistor. The output voltage (Vout) is proportional to the ratio of the feedback resistance to the resistance connected to the V_{ref}.

If we scale the input resistors with a **binary** *weighting* factor, each input can be made to *provide* a binary *weighted* amount of current, and the output voltage will represent a sum of all the binary-*weighted* input currents.

In Figure 6 the 20kΩ sums the currents that are *provided* by closing any of the switches D_0 to D_3.The resistors are scaled in such a way as to *provide* a binary-*weighted* amount of current to be summed by the 20kΩ resistor. Closing D_0 causes 50 μA to flow through the 20kΩ resistor, creating -1 at V_{out}. Closing each *successive* switch creates double the amount of current of the *previous* switch.

If we were to expand Figure 6 to an 8-bit D/A *converter*, the resistor for D_4 would be half of 12.5kΩ, which is 6.25kΩ. Each *successive* resistor is half of the *previous* one. Using this procedure, the resistor for D_7 would be 0.78125kΩ! Coming up with *accurate* resistance over such a large range of values is very difficult. This limits the practical use of this type of D/A *converter* for any more than 4-bit conversion.

Example:

Determine the voltage at V_{out} in Figure 6 if the binary *equivalent* of 10_{10} is input on switches D_3 to D_0.

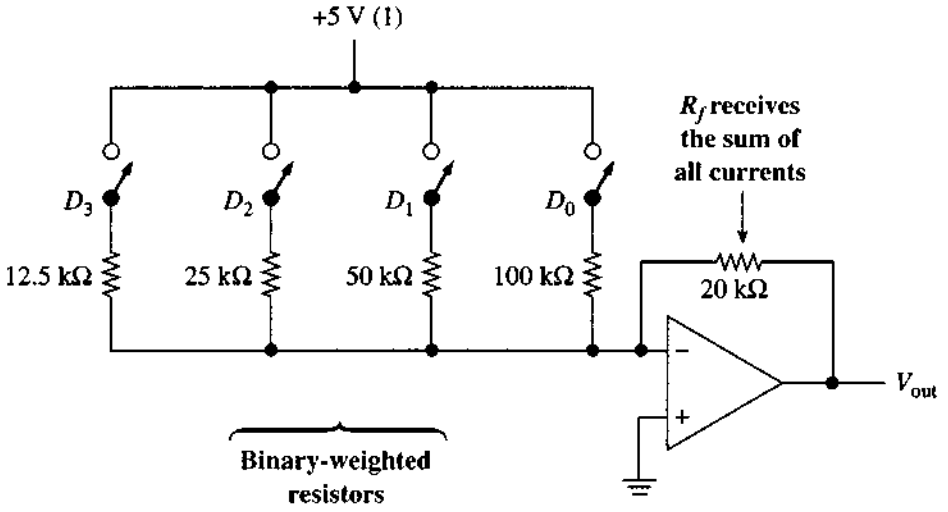
Solution: $10_{10} = 1010_2$ (switches D_3 and D_1 are closed)

$$I_3 = \frac{5V}{12.5k\Omega} = 0.4mA$$

$$I_1 = \frac{5V}{50k\Omega} = 0.1mA$$

$$V_{out} = -[(0.4mA + 0.1mA) \times 20k\Omega] = -10V$$

Figure 6: Binary Weighted Ladder

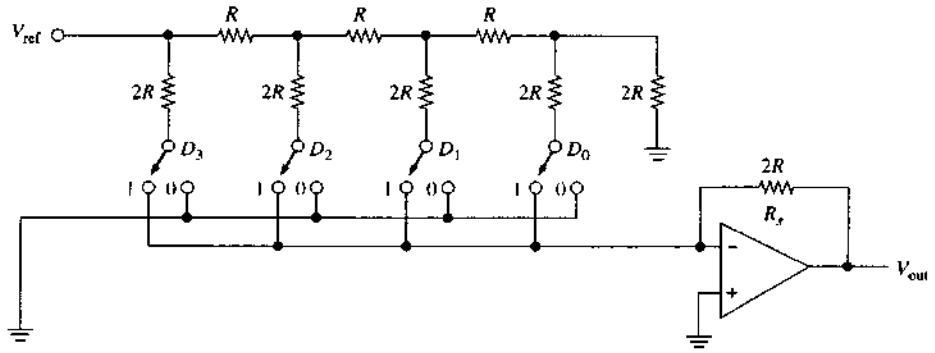


R/2R Ladder Digital-to-Analog Converters

The method for D/A conversion that is most often used in *integrated-circuit* D/A *converters* is known as the R/2R *ladder* circuit. In this circuit, only two resistor values are required, which *lends* itself nicely to the fabrication of *IC's* with a *resolution* of 8, 10, or 12 bits, and higher. Figure 7 shows a 4-bit D/A R/2R *converter*.

To form *converters* with higher *resolution*, all that needs to be done is to add more R/2R resistors and switches to the left of D_3. Commercially available D/ A *converters* with *resolutions* of 8, 10, and 12 bits are commonly made this way.

Figure 7: The R/2R ladder D/A converter.



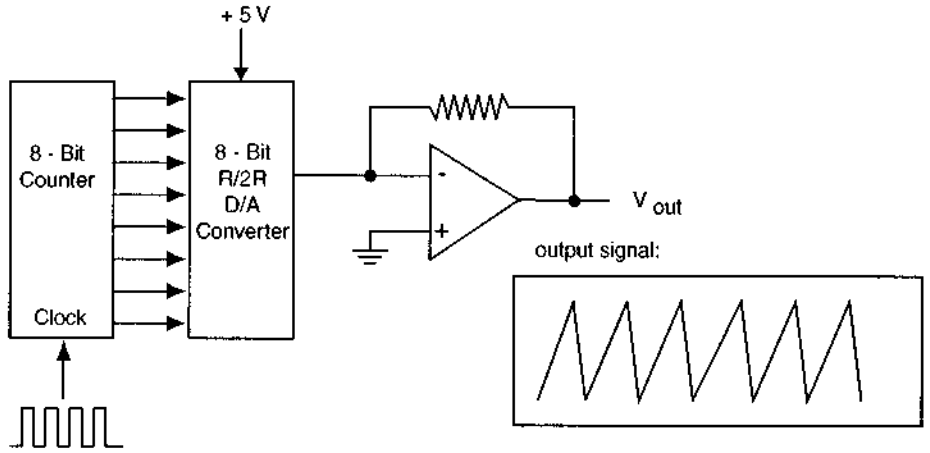
In Figure 7, the 4-bit digital information to be *converted* to analog is entered on the D_0 to D_3 switches. (In an actual *IC*, those switches would be transistor switches.) The arrangement of the circuit is such that as the switches are moved to the 1 position, they cause a current to flow through the summing resistor, Rs, that is proportional to their binary *equivalent* value. (Each *successive* switch is worth double the *previous* one.)

Waveform Generators

The analog output required from a digital computer is not always a steady output-*level*. Sometimes a particular wave form is required, such as a *sawtooth* or ramp. A ramp or *sawtooth* wave can be easily *implemented* by using a simple R-2R *ladder* and a counter.

A binary counter used to drive a R-2R *Ladder* causes the *ladder* to output a sequence of steps of different voltage *Levels*. As the counter reaches its maximum value, it returns to zero or is said to "rollover". When the input word rolls over to zero the output also returns to zero. While this output is not a pure ramp due to the fact that the binary words generate steps, most analog systems are slow enough in their reaction times that they react the same as if the signal was continuous. High quality R-2R ladders have relatively short response times (2 microseconds typical). For an 8 bit device this makes the rise time from 0 to 10 volts in the order of 0.5 microseconds or a frequency of 2 MHz.

Figure 8: Waveform Generator



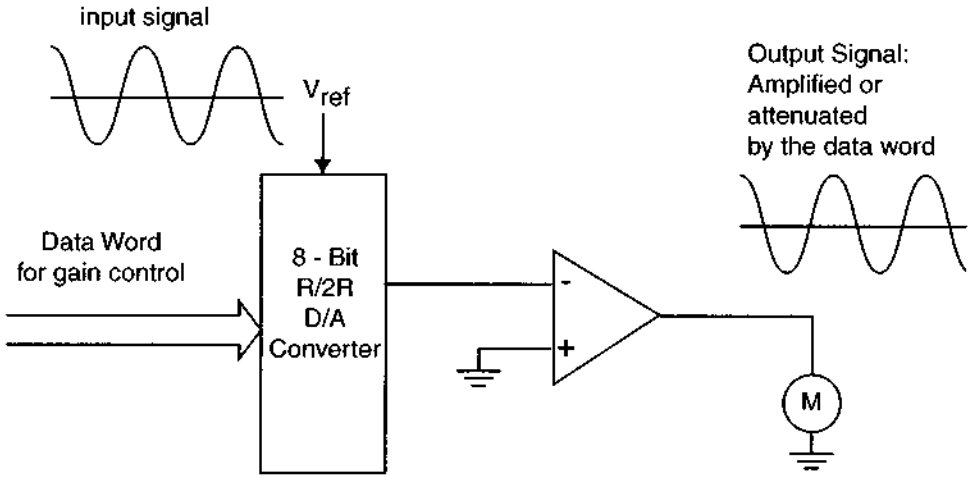
Programmable Gain Amplifiers

D/A *converters* can also be used to *provide gain* control of an analog signal. This arrangement may be required, for example, to control the speed of an AC motor by *varying* the input voltage *level* to the motor. This can be *accomplished* quite simply by *applying* an AC voltage as V_{ref} to a *ladder* network.

A simple motor controller is shown. A 5 volt AC reference signal is applied as V_{ref} to an 8-bit D/A *converter*. The data word is generated by the computer to select the desired motor speed by *setting* the output *level*. Remember for an 8-bit device this output can be divided into 256 steps, so very fine control can be *established* with this simple method.

One minor problem with this simple motor control is the fact that the computer does not know if the motor speed is correct. To check this, there must be a feedback from the motor. This feedback would be analog, therefore we need an analog to digital *converter*.

Figure 9: Programmable Gain Amplifier



Analog to Digital Converters

The *process* of taking an analog voltage and *converting* it to a digital signal can be done in several ways. One simple way that is easy to visualize is by means of parallel *encoding* (also known as simultaneous, multiple *comparator*, or flash *converting*). In this method, several *comparators* are set up, each at a different voltage reference *level* with their outputs driving a priority encoder as shown in Figure 10.

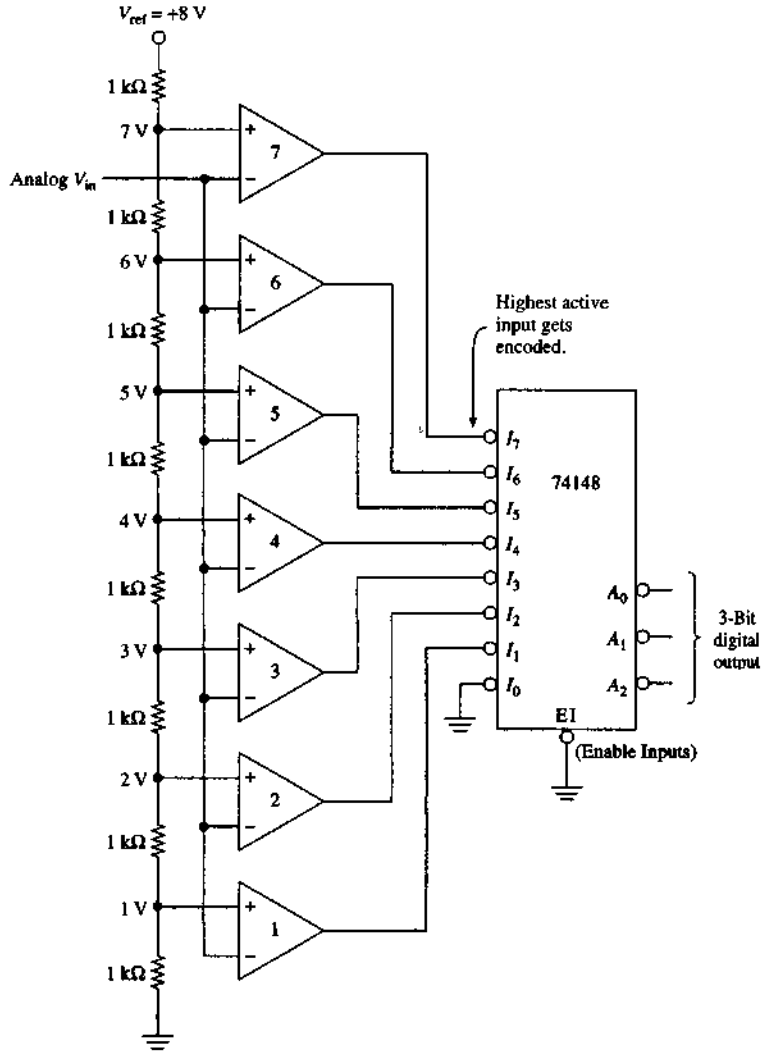
The voltage-divider network in Figure 10 is designed to *drop* 1 V across each resistor. This sets up a voltage reference at each *comparator* input in 1 V steps.

When V_{in} is 0 V, the + input on all seven comparators will be higher than the - input, so they will all output a HIGH. In this case, $\overline{I_0}$ is the only active-LOW input that is enabled, so the 74148 will output an active-LOW binary 0 (111).

When V_{in} exceeds 1.0 V, *comparator* 1 will output a LOW. Now $\overline{I_0}$ and $\overline{I_1}$; are both enabled, but because it is a priority encoder, the output will be a binary 1 (110).

As V_{in} increases further, each *successive comparator* outputs a LOW. The highest input that receives a LOW is encoded into its binary *equivalent* output.The A/D *converter* in Figure 10 is set up to *convert* analog voltages in the range from 0 to 7V. The range can be scaled higher or lower, depending on the input voltage *levels* that are expected. The *resolution* of this *converter* is only 3 bits, so it can only *distinguish* among eight different analog input *levels*. To expand to 4-bit *resolution*, eight more comparators are required to *differentiate* the 16 different voltage *levels*. To expand to 8-bit *resolution*, 256 *comparators* would be required! As you can see, circuit complexity becomes a real problem when using parallel *encoding* for high *resolution* conversion. However, a big advantage of using parallel *encoding* is its high speed. The conversion speed is limited only by the *propagation* delays of the comparators and encoder (less than 20 ns total).

Figure 10: Three-bit parallel-encoded A/D Converter



Counter-Ramp Analog-to-Digital Converters

The counter-ramp method of A/D conversion (ADC) uses a counter in *conjunction* with a D/A *converter* (DAC) to *determine* a digital output that is *equivalent* to the unknown analog input voltage. In Figure 11, depressing the start conversion push button clears the counter outputs to 0, which sets the DAC output to 0 V. The (-) input to the *comparator* is now 0 V, which is less than the positive analog input voltage at the (+) input. Therefore, the *comparator* outputs a HIGH, which enables the AND gate, allowing the counter to start counting. As the counter's binary output increases, so does the DAC output voltage in the form of a *staircase.*

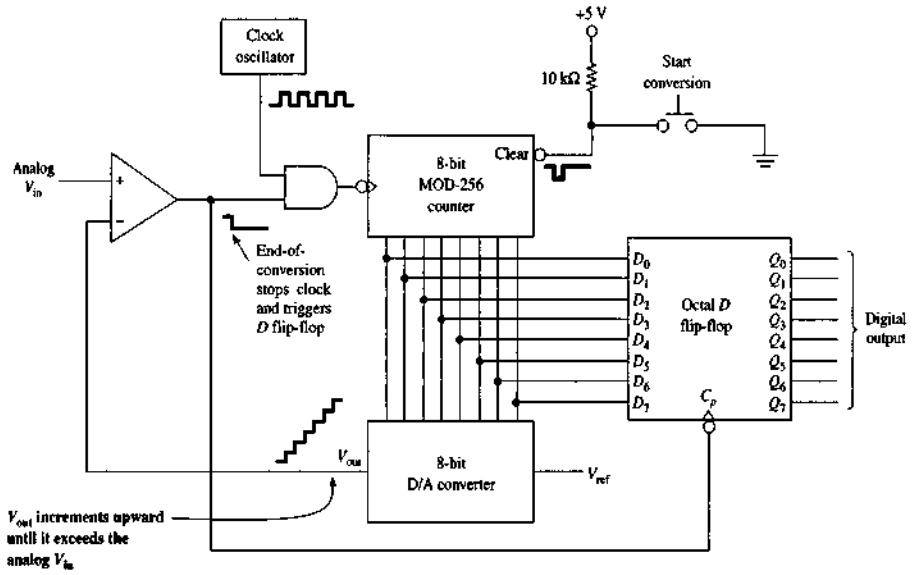
When the *staircase* voltage reaches and then exceeds the analog input voltage, the *comparator* output goes LOW, *disabling* the clock and stopping the counter. The counter output at that point is equal to the binary number that caused the DAC to output a voltage slightly greater than the analog input voltage. Thus we have the binary *equivalent* of the analog voltage!

The HIGH-to-LOW *transition* of the *comparator* is also used to *trigger* the D *flip-flop* to latch on to the binary number at that instant. To perform another conversion, the start push button is depressed again and the *process* repeats. The result from the *previous* conversion remains in the D *flip-flop* until the next end-of-conversion HIGH-to-LOW *edge* comes along.

To change the circuit to perform continuous conversions, the end-of -conversion line could be tied back to the clear input of the counter. A short *delay* needs to be inserted into this new line, however, to allow the D *flip-flop* to read the binary number before the counter is reset. Two inverters placed end to end in the line will produce a sufficient *delay*.

The main disadvantage of the counter-ramp method of conversion is its slow conversion speed. The worst-case maximum conversion time will *occur* when the counter has to count all 255 steps before the DAC output voltage matches the analog input voltage.

Figure 11: Counter-Ramp Analog-to-Digital Converter



Successive-Approximation Analog-to-Digital Conversion

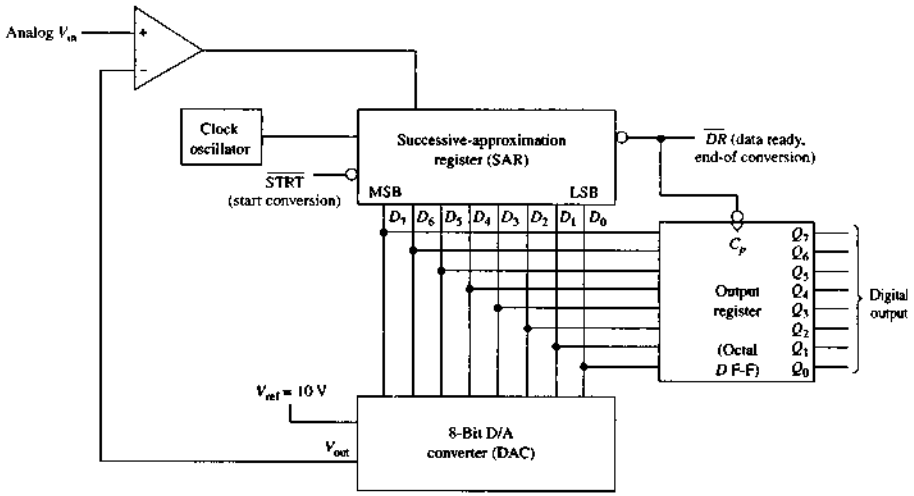
Other methods of A/D conversion *employ* up/down-counters and *integrating slope converters* to *track* the analog input, but the method used in most modem *integrated-circuit* A/DC's is called *successive approximation*. This *converter* circuit is similar to the counter-ramp ADC circuit except that the method of narrowing in on the unknown analog input voltage is much *improved*. *Instead* of counting up from "0" and comparing the DAC output each step of the way, a *successive-approximation* register (SAR) is used in place of the counter (Figure 12).

In Figure 12 the conversion is started by *dropping* the STRT line LOW. Then the SAR first tries a HIGH on the *MSB* (D_7) line to the DAC. (Remember, D_7 will cause the DAC to output half of its full-scale output.) If the DAC output is then higher than the unknown analog input voltage, the SAR returns the *MSB* LOW. If the DAC output was still lower than the unknown analog input voltage, the SAR leaves the *MSB* HIGH.

Now, the next lower bit (D_6) is tried. If a HIGH on D_6 causes the DAC output to be higher than the analog V_{in}, it is returned LOW. If not, it is left HIGH. The *process* continues until all 8 bits, down to the *LSB*, have been tried. At the end of this eight-step conversion *process*, the SAR *contains* a valid 8-bit binary output code that represents the unknown analog input. The $\overline{DR}$ output now goes LOW, indicating that the conversion is complete and the data are ready. The HIGH-to-LOW *edge* on $\overline{DR}$ clocks the D_0 to D_7 data into the octal D flip- flop to make the digital output results available at the Q_0 to Q_7 lines.

The main advantage of the SAR ADC method is its high speed. The ADC in Figure 12 takes only eight clock periods to complete a conversion, which is a *vast improvement* over the counter-ramp method.

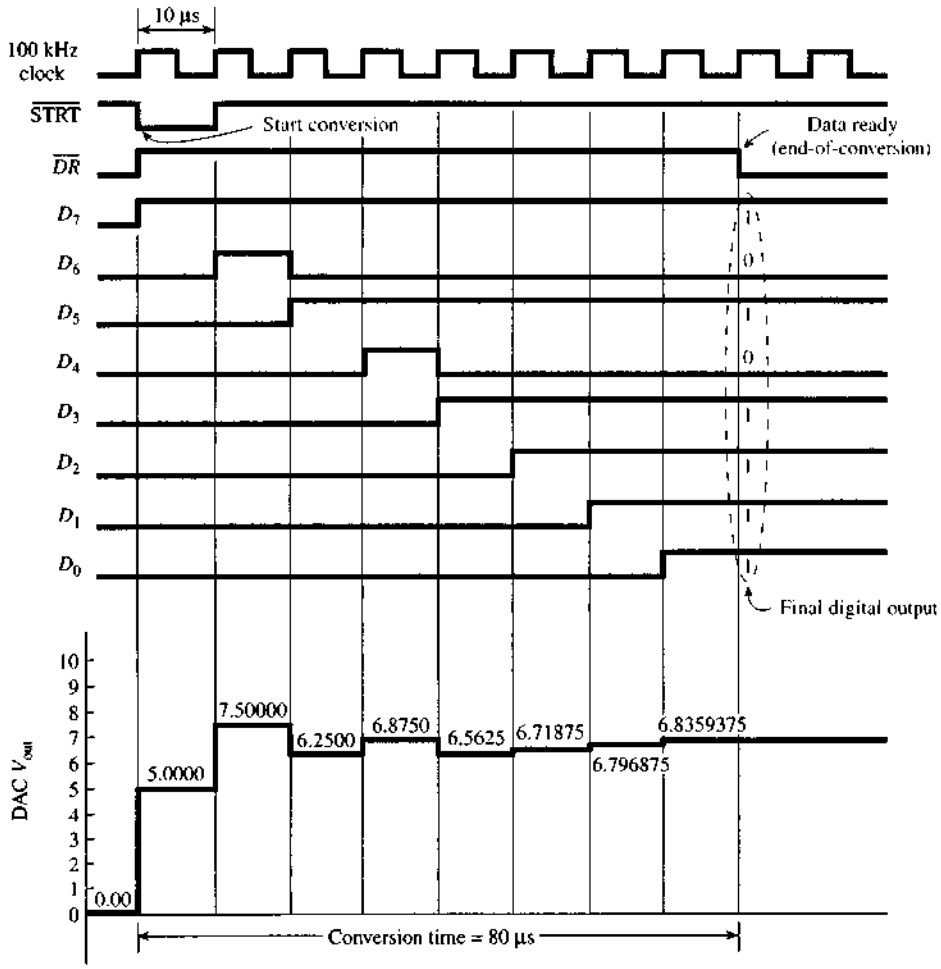
Figure 12: Successive-Approximation Analog-to-Digital Conversion



Timing Waveforms for a Successive Approximation A/D Conversion

To watch the conversion in progress an eight-channel oscilloscope or logic analyzer can be connected to the D_0 to D_7 outputs of the SAR. For continuous conversions the $\overline{DR}$ line can be connected back to the $\overline{STRT}$ line. That way, as soon as the conversion is complete, the HIGH-to-LOW on DR will issue another start conversion ($\overline{STRT}$), which forces the data ready ($\overline{DR}$) line back HIGH for eight clock periods while the new conversion is being made. The latched Q_0 to Q_7 digital outputs will always display the results of the *previous* conversion.

Figure 13: Timing waveforms for a successive approximation A/D conversion



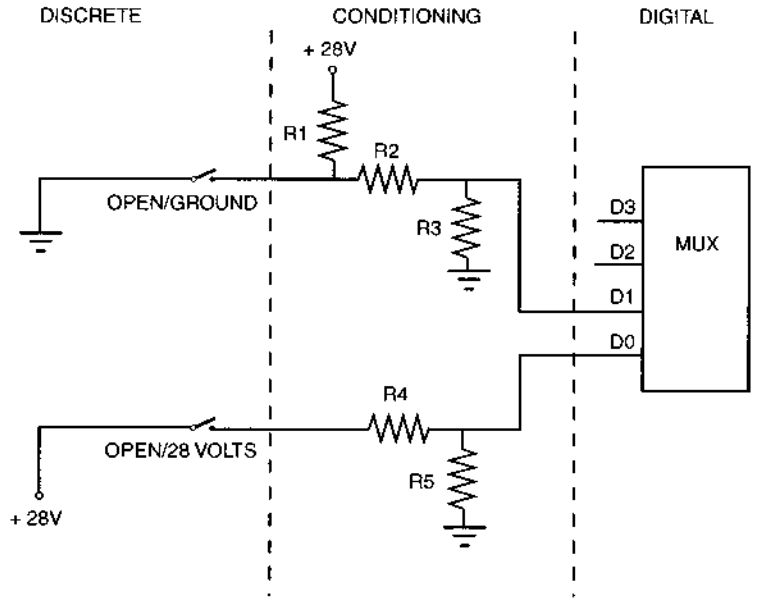
Discrete Conditioning

Many times *discrete* inputs are at voltage *levels* which are incompatible with the digital system in use. Most digital systems *recognize* a logic "0" as ground or some voltage range near 0 volts and logic "1" as 5 volts or some voltage range near 5 volts. As such, the common *discrete* forms of open/ground and open/28 volts require conditioning. This conditioning is typically performed by a simple resistor network.

An open/ground *discrete* input utilizes a resistive divider network connected from +28 volts (or some *appropriate* positive voltage) to ground. The resistor values of R1, R2, and R3 are selected to *provide* a logic "1" *level* voltage when the *discrete* is open and a ground when the *discrete* is ground. An open/28 volt *discrete* input is held at ground (logic "0") in the *discrete* open position. Resistors R4 and R5 *provide* a logic "1" *level* voltage to the digital system in the *discrete* 28 volt position. *Discrete* inputs of this type are typically *buffered* or *multiplexed* to the digital system.

The Resistors, in this case R1 and R5, that pull the voltage up to the correct *level*, or down to ground, can also be referred to as **pull up**, and **pull down resistors**.

Figure 14: Discrete Conditioning



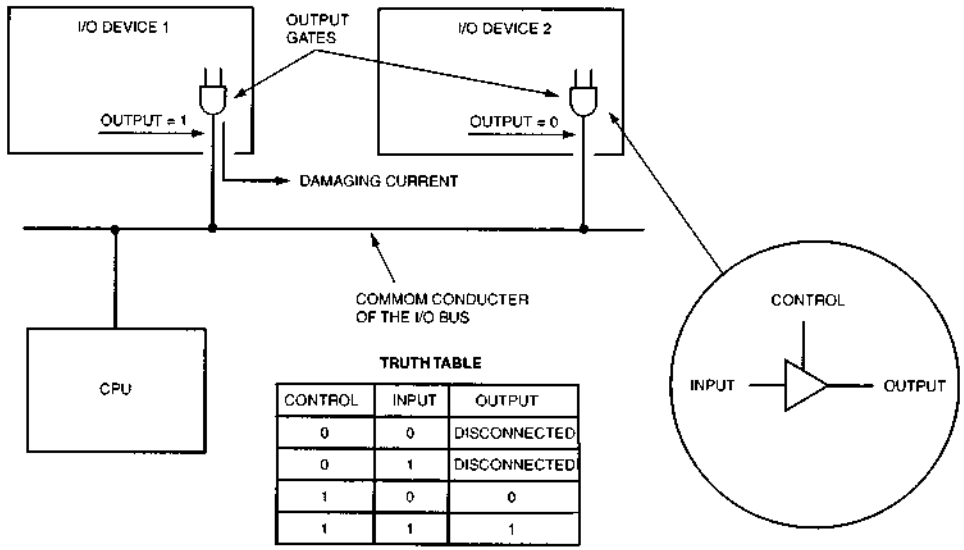
Tristate Buffers

Since the output of most logic circuits is a low *impedance*, they cannot be connected together to the same line, such as a data bus, because of the damaging currents that result from the voltage difference when one such device is logic high and the other device is a logic low. This situation is avoided by use of the *tristate buffer*. This device has three possible output *states*: 1, 0, or high- *impedance*.

The high *impedance* (or *floating*) *state* essentially disconnects the output from the input when a logic "0" is applied to the control input. A logic "1" applied to the control gate allows the output of the device to be logic "0" or "1" in response to its input. In this *state* the *tristate buffer* electrically "looks" like a standard logic device. As such, care is always taken in a design to assure that only one *buffer* on any given bus be turned on at one time.

*Tristate buffer*s are made using various logic gate configurations (AND's, OR's, INVERTERS) required by a given system. They are also generally packaged as multiple devices to *accommodate* parallel data.

Figure 15: Tristate Buffers



TRUTH TABLE

CONTROL	INPUT	OUTPUT
0	0	DISCONNECTED
0	1	DISCONNECTED
1	0	0
1	1	1

5.4 Data Buses

Introduction

From the computers within the aircraft systems, information has to be transmitted to the components, this may be in form of electrical or optical signals and the medium for the transmission is known as data buses.

In the first part of this chapter, a general overview about data transmission is given. In the second part we will focus on the data bus structure used in aviation technology.

Data Transmission in General

Designing an interface, or simply using an existing interface, to connect two devices involves a number of issues. For example, digital interfacing can be categorized as parallel or serial, internal or external and asynchronous or synchronous. Additional issues are the data rate, error *detection* methods and the signalling format or standards. The format can be especially important since many standards and conventions have developed that should be taken into *consideration*. This first part of the chapter focuses on some basic concepts of digital communications for interfacing between devices.

Parallel Versus Serial Signalling

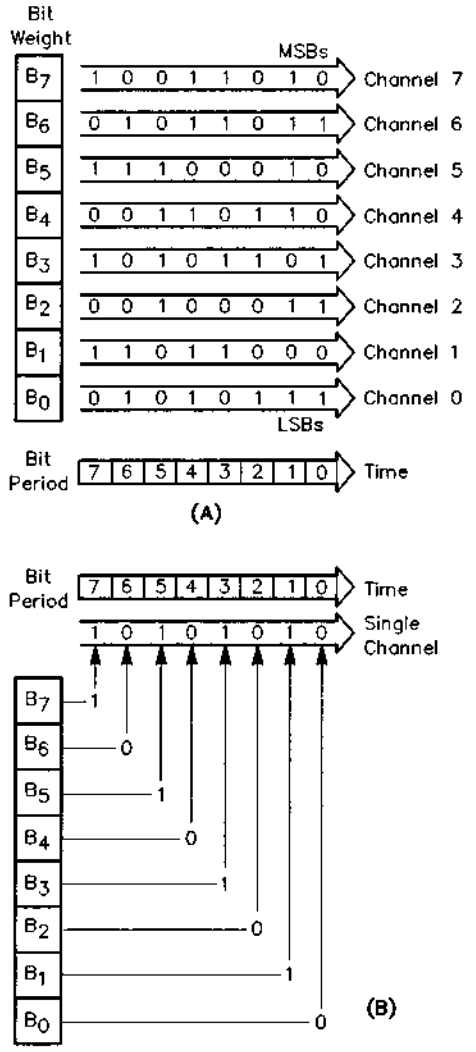
To communicate a word to you across the room, you could hold up flash cards displaying the letters of the word. If you hold up four flash cards, each with a letter on it, all at once, then you are transmitting in parallel. If *instead*, you hold up each of the flash-cards only one at a time, then you are transmitting in serial. Parallel means all the bits in a group are handled exactly at the same time. Serial means each of the bits is sent in turn over a single channel or wire, according to an agreed sequence. Figure 1 gives a graphic illustration of parallel and serial signalling.

Both parallel and serial signalling are *appropriate* for certain *circumstances*. Parallel signalling is faster, since all bits are transmitted *simultaneously*; but each bit needs its own *conductor*, which can be expensive. Parallel signalling is more likely to be used on internal communications. For longer distance communications, such as to an external device, serial signalling is more *appropriate*.

Each bit is sent in turn, so communication is slower; but it is also less expensive, since fewer channels are needed between the devices.

Most avionics applications use serial transmission, to minimize *weight* and complexity. The number of channels needed for parallel or serial signalling also depends on the operational mode: one channel per bit for simplex (one-way, from sender to receiver only) and for half-duplex (two-way communication, but only one-way transmission is possible at a time) but two channels per bit for full-duplex (simultaneous communications in both directions).

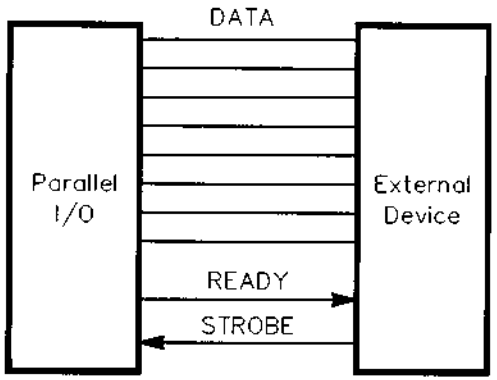
Figure 1: Parallel Versus Serial Signalling



Parallel I/O Interfacing

Typically, they have eight data lines and one or more handshaking lines. *Handshaking* involves a number of functions to coordinate the data transfer. For example, the READY line indicates that data is available on all 8 data lines. If only the READY line is used, however, the receiver may not be able to keep up with the data. Thus, the STROBE line is added so the receiver can watch to ensure the transmitter is ready for the next character

Figure 2: Parallel I/O Interfacing.



Serial I/O Interfacing

Serial input/output interfacing is more complex than parallel, since the data must be transmitted *based* on an agreed sequence. For example, transmitting the 8 bits (b7, b6,... b0) of a word includes specifying whether the least significant bit, b0, or the most significant bit, b7, is sent first. Fortunately, a number of standards have developed to define the agreed sequence, or *encoding* scheme.

Conversions

Within computers and other digital circuits, data is usually operated on, stored and transmitted in parallel. For communicating with an external device, data must usually be converted from parallel to serial format and vice versa. This conversion is usually handled by shift registers.

Shift registers can be left shifters, right shifters or controlled to shift in either direction. The most general form, a universal shift register, has two control inputs for four states: Hold, Shift right, Shift left and Load. Most also have asynchronous inputs for preset, clear and parallel load.

Asynchronous versus Synchronous Communication

To correctly receive data, the receiving interface must know when data bits will *occur*; it must be synchronized with the sender. In asynchronous communication, the receiver synchronizes on each incoming character. Each character includes start and stop bits to indicate the beginning and end of that character. In synchronous communication, data is sent in long blocks, without start and stop bits or *gaps* between characters.

Asynchronous Communication

In asynchronous communication, each transmitted character begins with a start bit and ends with a stop bit, as shown in Figure 3. The start bit (usually a zero) tells the receiver to begin receiving a character. The stop bit (usually a one) signals the end of a character. Between characters, the transmitting circuit sends the stop bit state (steady one or zero).

Since the receiver is always told when a character begins and ends, characters can be sent at irregular intervals. This is especially advantageous for typed input, since the person typing is usually slower than the data communications equipment and will usually work at an uneven pace. An other advantage of asynchronous data is that it does not need complex circuits to keep it synchronized. Since the receiver is newly synchronized at the beginning of each character, the characters need not be sent in a steady stream and no *stringent* demands are made on the person or *process* generating the characters.

A disadvantage of asynchronous communications is the inclusion of the start and stop bits, which are not "useful" data. If you are transmitting 8 data bits, 1 start bit and 1 stop bit, then 20% (2 of 10 bits) is overhead.

Synchronous Communication

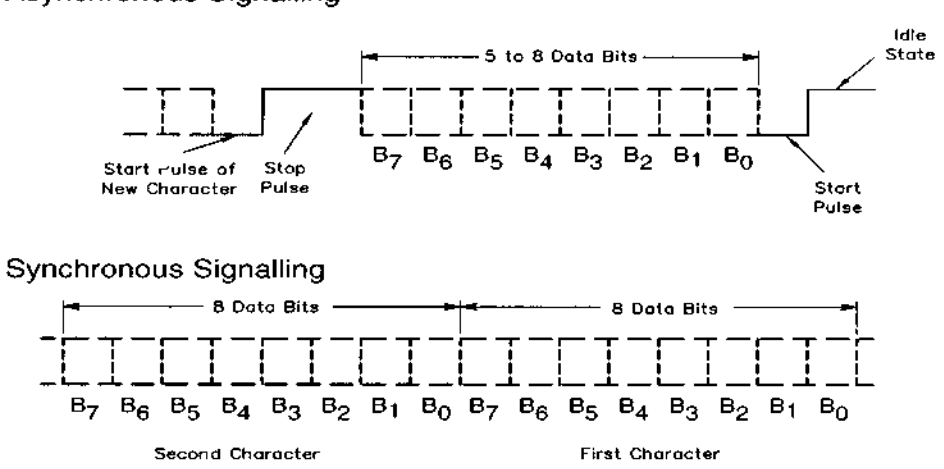
In synchronous communications, data is sent in blocks, usually longer than a single character, as shown in Figure 3. At the beginning of each block, the sender transmits a special sequence of bits that the receiver uses for initial synchronization. After becoming synchronized at the beginning of a block, the receiver must stay synchronized throughout the block. The sender and receiver may be using slightly different clock frequencies, so it is usually not *adequate* for them to *merely* be synchronized initially. There are several ways for the receiver to stay synchro-

nized. The transmitter may send the clock signal on a separate channel, but this is wasteful.

One disadvantage of synchronous transmission is that the data must be sent as a continuous stream; characters must be placed in a *buffer* until there are enough to make a block.

The major advantage of synchronous signalling is that it does not impose overhead (the start and stop bits) on each character. This is an important *consideration* during large data transfers.

Figure 3: Serial data transmission format.



Data Rate

Within a given piece of equipment, it is desirable to use the highest possible data rate. When external devices are interfaced, it is normal practice to select the highest standard signalling rate at which both the sending and receiving equipment can operate.

There are a number of limitations on how fast data can be transferred:

- The sending equipment has an upper limit on how fast it can produce a continuous stream of data.
- The receiving equipment has an upper limit on how fast it can accept and *process* data.
- The signalling channel itself has a speed limit, often based on how fast data can be sent without errors.
- Finally, standards and the need for *compatibility* with other equipment may have a strong influence on the data rate.

Two ways to express data transmission rates are baud and bits per second (bps). These two terms are not interchangeable:

Baud rate is the *reciprocal* of the shortest signal element (a measure of the number of line changes which *occur* every second). For a binary signal of 20Hz, this is *equivalent* to 20 baud (there are 20 changes per second).

Bits Per Second is an expression of the number of bits per second. Where a binary signal is being used, this is the same as the baud rate. When the signal is changed to another form, it will not be equal to the baud rate, as each line change can represent more than one bit (either two or four bits).

The maximum information transfer rate is defined as the number of *equivalent* binary digits transferred per second; this is measured in bits per second.

Error Detection

Since data transfers are subject to errors, data transmission should include some method of detecting and correcting errors. Numerous techniques are available, each used depending on the specific *circumstances*, such as what types of errors are likely to be encountered. One of the simplest and most common techniques is the *parity* check.

Parity Check

Parity check *provides adequate* error *detection* for some data transfers. This method transmits a *parity* bit along with the data bits. In systems using odd *parity*, the *parity* bit is selected such that the number of 1 bits in the transmitted character (data bits plus *parity* bit) is odd. In even *parity* systems, the *parity* bit is chosen to give the character an even number of ones.

For example, if the data **1101001** is to be transmitted, there are 4 (an even number) ones in the data. Thus, the *parity* bit should be set to 1 for odd *parity* (to give a total of 5 ones) or should be 0 for even *parity* (to maintain the even number, 4). When a character is received, the receiver checks *parity* by counting the ones in the character. If the *parity* is correct, the data is *assumed* to be correct. If the *parity* is wrong, an error has been detected.

Parity checking only detects a small fraction of possible errors. This can be intuitively understood by noting that a *randomly* chosen word has a 50% chance of having even *parity* and a 50% chance of having odd *parity*. Fortunately, on relatively error-free channels, single-bit errors are the most common and *parity* checking will always detect a single bit in error. *Parity* checking is a simple error *detection* strategy. Because it is easy to *implement*, it is frequently used.

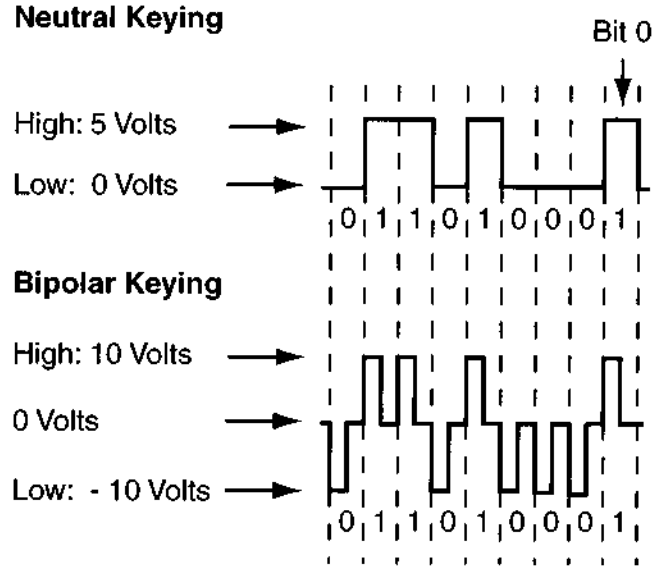
Signalling Levels

Inside equipment and for short runs of wire between equipment, the normal practice is to use neutral keying; that is, simply to key a voltage such as + 5 V on and off. In neutral keying, the off condition is *considered* to be 0 V. Over longer runs of wire, the line is viewed as a transmission line, with distributed *inductance* and *capacitance*. It takes longer to make the *transition* from 0 to 1 or vice versa because of the additional *inductance* and *capacitance*. This decreases the maximum speed at which data can be transferred on the wire and may also cause the 1s and 0s to be different lengths, called *bias distortion*. Also, longer lines are more likely to pick up *noise*, which can make it difficult for the receiver to decide exactly when the *transition* takes place.

Because of these problems, polar keying (technically bipolar keying) is used on longer lines. Polar keying uses one polarity (for example +) for a logical 1 and the other (– in this example) for a 0. This means that the decision *threshold* at the receiver is 0 V. Any positive voltage is taken as a 1 and any negative voltage as a 0. As the voltage level returns to 0V after each signal transmission, this type of keying is also called **"Bipolar Return To Zero"**.

Since neutral keying is usually used inside equipment and polar keying for lines leaving the equipment, signals must be converted between polar and neutral. Op amp circuits, line drivers and line receivers are ways to handle this conversion.

Figure 4: Neutral Keying vs. Bipolar keying



ARINC

Aeronautical Radio Incorporated (ARINC) is a corporation made up of scheduled airlines, transport companies, aircraft manufactures and foreign flag airlines.

One primary activity of ARINC is to produce specifications and reports for the purpose of:

- Indicating to manufacturers the group opinion concerning requisites of new equipment
- To channel new equipment designs in a direction which will result in maximum standardisation

.ARINC Standards are publications that define the avionics systems and equipment used in most transport category aircraft. The standards *provide* a common basis for physical, electrical, and functional interface design. They allow equipment and airframe designs to be coordinated early in the pre-production stages.

The following definition is given by ARINC for the different series of ARINC Standards:

700 Series

ARINC Standards in the 700 Series define digital avionics systems and equipment installed on current model production aircraft. These Standards include detailed definitions of form, fit, function, and interface.

600 Series

ARINC Standards in the 600 Series *provide* a *foundation* for avionics equipment built according to specifications in the ARINC 700 Series. These Standards include provisions for installation and mounting, data transfer, test, and maintenance. Data link protocol standards are also defined.

500 Series

ARINC Standards in the 500 Series define older analog avionics equipment used *widely* on the B-727, DC-9, and DC-10 and on early models of B-737, B-747, and A-300 aircraft.

400 Series

ARINC Standards in the 400 Series *provide* a *foundation* for avionics built according to specifications in the ARINC 500 and ARINC 700 Series. Included are installation, wiring, data buses, data bases, and general *guidance*.

In the following section, the most common used Bus in aircraft communication technology, ARINC 429 will be studied in detail. The next generation data bus standard ARINC 629 will also be explained. For other data bus standards, refer to the specific standard manuals.

The ARINC 429 System

Characteristics of ARINC 429

Serial transmission of information in digital format as defined by ARINC specification 429. System data transfers in and out of an aircraft system and within the system *occur* in digital formats as defined by ARINC specification 429 Mark 33 Digital Information Transfer System (DITS).

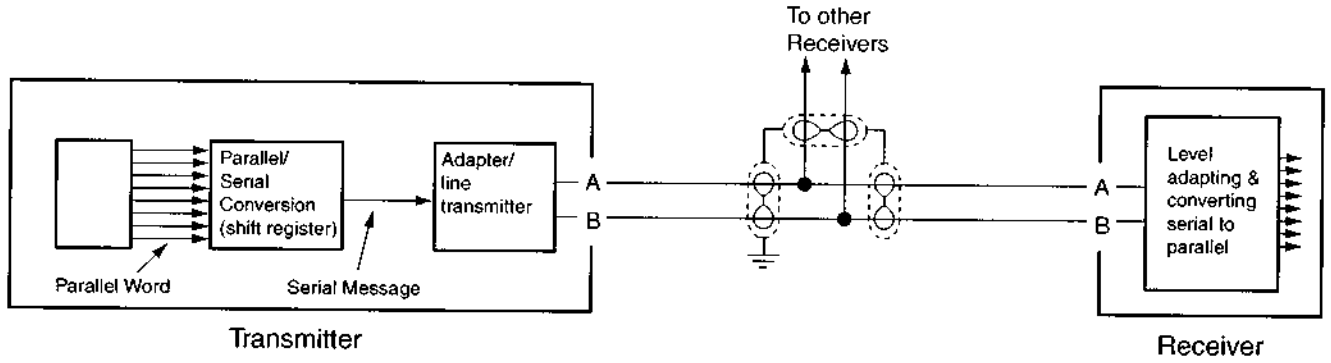
The characteristic ARINC 429 covers hardware and protocol for communicating on the ARINC 429 bus. Today, in local area network, LAN, terms, we would say ARINC 429 describes the physical *layer* and data link *layer* of the ARINC 429 LAN. When ARINC 429 was conceived in the mid 1970's there were few LAN's and these terms were not known.

The digital computers of the different aircraft systems, *process* results in the form of messages or parallel binary words, e.g., information comprising several bits (0,1) available *simultaneously*. However, to transmit digital information towards external receivers, it is preferable, for *weight gain* and *reliability* reasons, to use a serial transmission system. Therefore, the parallel message is converted into a serial message. Then a line transmitter adapts this serial logic message into voltage levels which are compatible with the transmission standard. The message is thus sent in the form of a *string* of pulses.

The hardware support providing serial transmission of information is a mono-directional bus composed of a pair of twisted and shielded wires. This shielding is connected to ground, in particular at each *branch*.

Figure 5 shows the architecture of the ARINC 429 transmission system.

Figure 5: ARINC Transmission System

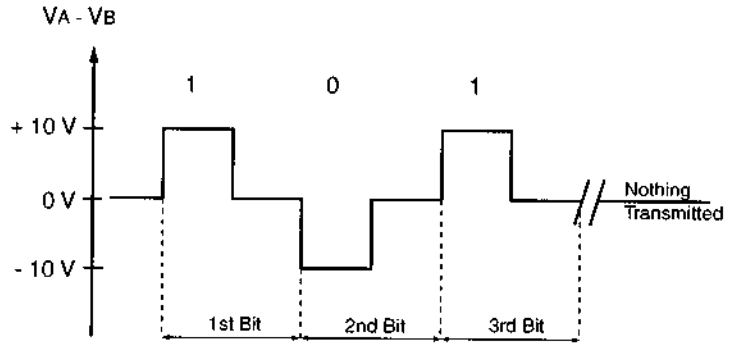


The ARINC 429 Digital Information Transfer System (DITS) is a self-clocking, self-synchronising system called "bipolar return to zero". See Figure 4

The *successive* bits which form a word are represented in positive logic. A differential voltage between wire A and wire B of + 10 Volts represents a 1 bit. If this voltage is -10 Volts, a 0 bit is transmitted. If the differential voltage remains at 0 Volts, nothing is transmitted.

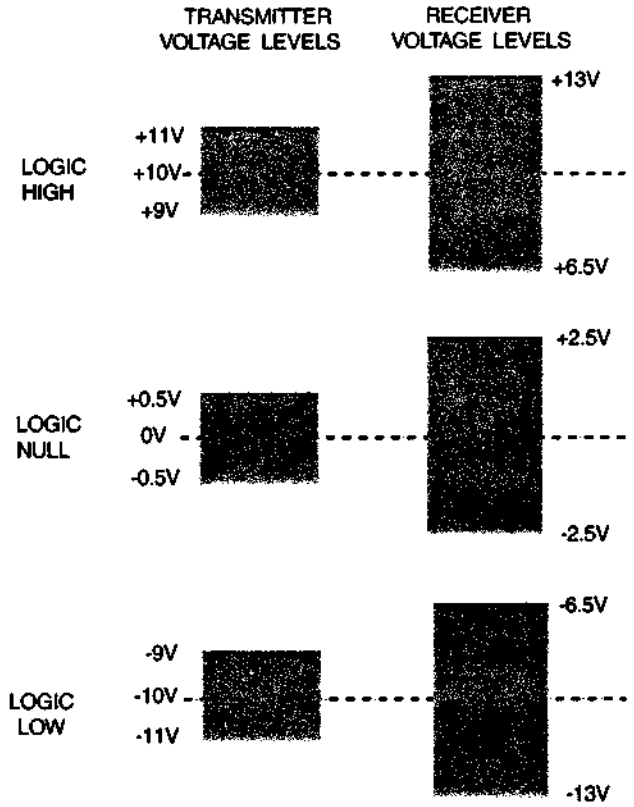
The data is transmitted in groups of 32 bits in serial fashion i.e. one bit at a time. Signals only flow in one direction on the bus.

Figure 6: Voltage Levels ARINC 429



Transmission and reception tolerances are given for the voltage levels.

Figure 7: Voltage Tolerances



Other characteristics of an ARINC 429 bus link:

Rate at which bits are transmitted:

- The bit rate for low speed operation of the system should be within the range 12.0 to 14.5 kilobits per second. The selected rate should be maintained within 1%.
- The bit rate for high speed operation of the system should be 100 kilobits per second ±1%.

So, the time during which a given bit of a word occupies the lines is approximately 80 µsec, at slow speed and 10 µsec at fast speed.

The beginning of a word is detected by the first ascending leading *edge*. (for a 1 bit) or descending leading *edge* (for a 0 bit).

With this type of transmission, there is no need for a synchronization signal or an additional clock signal, sent on another wire and used to read the bits received in the receiver correctly, at the right moment.

The *successive* words transmitted along the same wire are spaced by intervals of time *corresponding* to the emission of 4 bits. Throughout this time, the differential voltage of the lines is 0V.

The transmitters of the system, which are *embedded* in the system equipment, are capable of interfacing with up to 20 receivers.

The ARINC 429 Data Words are 32 bit *wide* and have the five basic parts:

1. Label
2. Source/Destination Identifier (SDI)
3. Data Field
4. Sign Status Matrix (SSM)
5. *Parity* Bit (P)

In the following section, the significance of the different parts of an ARINC 429 word will be explained. Refer to Figure 8 for word structure and location of the different parts within the data words.

Figure 8: ARINC 429 General Word Formats and Encoding Examples

1. General Word Formats

32	31 30	29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11	10 9	8 7 6 5 4 3 2 1
P	SSM	DATA ——— MSB ——— PAD ——— DISCRETES ——— LSB	SDI	LABEL

Generalized BCD Word Format

P	SSM	BCD CH #2	BCD CH #2	BCD CH #3	BCD CH #4	BCD CH #5	SDI	8 7 6 5 4 3 2 1
		4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1		
0	0 0	0 1 0	0 1 0 1	0 1 1 1	1 0 0 0	0 1 1 0	0 0	1 0 0 0 0 0 0 1
Example		2	5	7	8	6		DME DISTANCE

BCD Word Format Example (No Discretes)

32	31 30	29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11	10 9	8 7 6 5 4 3 2 1
P	SSM	DATA ——— MSB ——— PAD ——— DISCRETES ——— LSB	SDI	LABEL

Generalized BNR Word Format

P	SSM	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{128}$	etc	PAD	SDI	LABEL
0	1 1 0	0	0	1	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0	0	0 0	0 1 1 0 1 1 1 1
Example		512 Knots (i.e. 1/8 x 4096 where 4096 is entry in range column of Table 2 in Attachment 2.										N-S VELOCITY

BNR Word Format Example (No Discretes)

Pad (comp)

All bit positions not used for data or discretes should be filled with binary zero or valid data pad bits.

ARINC Word Label

The Label, the first eight bits of the word, identifies the information *contained* in the data i.e. DME distance, N-S Velocity, exhaust gas temperature etc. There are a large numbers of words being transmitted on the bus and the receiver decodes the labels and selects only those words it requires.

Source/Destination Identifier

This is used to identify the source or destination of a word and is transmitted as bits 9 and 10 in the word i.e., which of a number of installations the word is coming from or needs to be directed to.

Data Field

The data field *contains* the specific data related to the label i.e., how many knots for *airspeed*, value of exhaust gas temperature etc. For a binary word this is *contained* in bits 11-28 and for a binary coded decimal (BCD) word it is in bits 11-29. Any bits not used are filled with logic 0's these are known as pad bits and have no data significance.

Sign Status Matrix

The Sign Status Matrix, bits 29,30,31 for a binary word and 30,31 for a BCD word identifies the characteristics of the word i.e., north or south, positive or negative, east or west and it's status i.e., no computed data, failure warning or functional test or normal operation.

Parity Bit

The *PARITY* BIT. ARINC 429 uses odd *parity* i.e. the total number of logic '1 's in the word must be an odd number, if it is not an odd number, the *parity* bit is set to 1. This is used in the system to check for errors, if on receiving a signal it does not *contain* an odd number of '1's then there is something wrong with the transmission and a fault signal would be generated.

MIL-STD-1553B

This is a United States Military Standard and has been adopted as a NATO standard (STANAG 3838). It is a multiple source data transmission system in that transmission can come from more than one source. It is a half duplex system in that the data transfer can take place in either direction on a single line but not in both directions on that line *simultaneously*.

Designed for the military, MIL STD 1553 is used in limited cases for civilian aircraft. However, only the newest aircraft use serial data transmission extensively. ARINC 629 is similar to MIL STD 1553 and is superior to ARINC 429. It is expected that ARINC 629 will become more *prevalent* in new aircraft than ARINC 429. MIL STD 1553 is also used in aircraft ground test equipment and simulation.

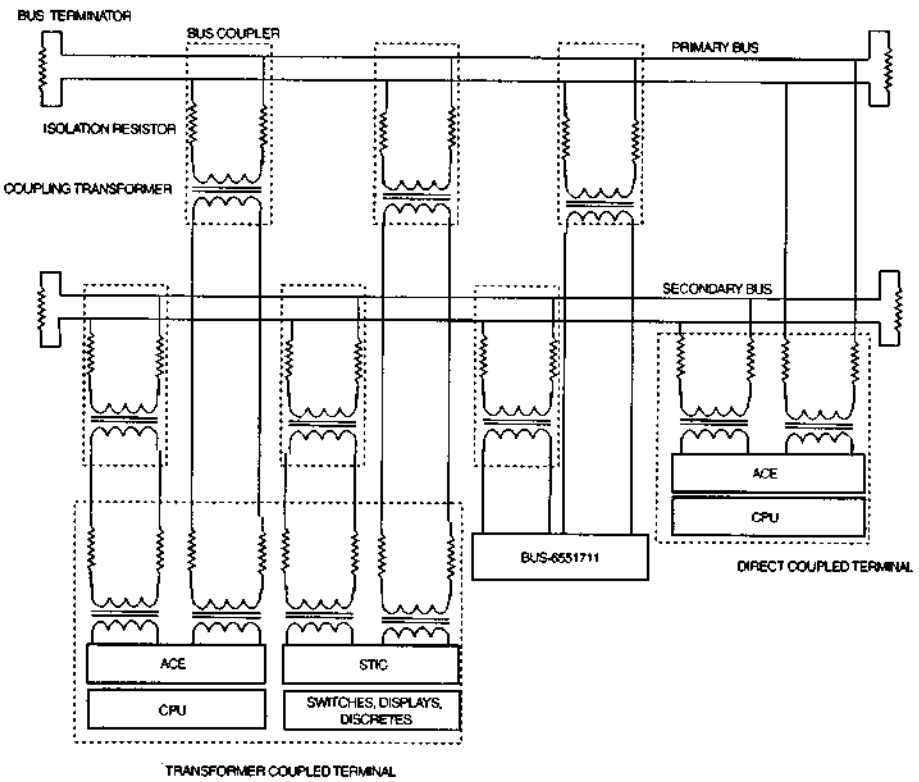
MIL STD 1553 is a higher speed and more versatile data communications system than ARINC 429. One of the first differences is the speed of MIL STD 1553, which is 1Mb/s. There is only one speed as there are no variants of the system.

The system operates with a shielded twisted pair using balanced differential signalling. This is common in advanced wired communications because, other than coaxial cable, anything less does not work.

Data is transmitted in asynchronous packets of 20 bits each. Unlike ARINC 429, however, a bus can have more than one transmitter. This difference significantly affects the topology of the communications system and *provides* for high *reliability*.

Figure 9 shows a MIL STD 1553 system. Subscribers to the bus are called remote terminals, or RT's. Like any terminal, RT's can receive and transmit. A bus controller reduces the number of collisions and other bus *contentions*. MIL STD 1553 *permits redundant* buses, a very important characteristic in a military system, which needs to function in spite of battle damage. In critical civilian applications, such as fly by wire, a similar advantage is obtained from redundancy.

Figure 9: MIL STD 1553 Communications System



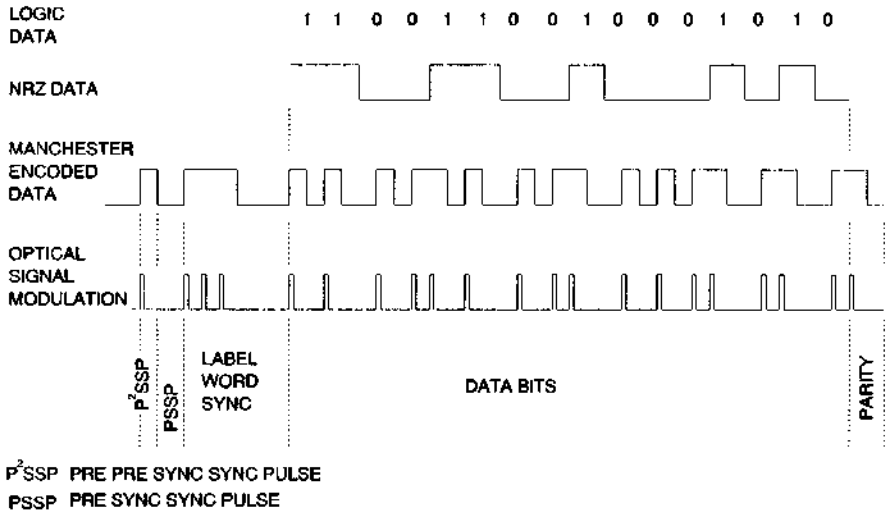
A bi-directional bus using RT's rather than a dedicated driver and only receivers significantly affects the way data is handled. First, in the single driver per bus, after data is transmitted, there is no acknowledgment of a successful receipt. The ability to perform two-way handshaking is an important part of a high *reliability* communications system.

In addition to improving data integrity, the bi-directional bus *permits considerable* flexibility and reduces the amount of twisted pair. A number of RT's with data to send and receive are present on the same bus. In the case of ARINC 429, each *provider* of data requires a bus for the *provider* and each recipient of data has a receiver on the bus.

The only limitation on the number of data *providers* on a MIL STD 1553 bus is speed. For slow speed data, a large number of data sources can reside on the bus, as the 1 Mb/s data rate results in short data words and reduces the likelihood of collisions.

Data bits are applied to the bus using bipolar Manchester *encoding*. Each data bit has a time slot and if the transmitted pulse is in the early part, the bit is a logic one. If the transmitted pulse is in the late part of the time slot, the transmitted bit is a logic zero. In the case of MIL STD 1553, the transmission is not of an RF *carrier* but voltage across a transmission line. For a logic one, a positive pulse followed by a negative pulse is placed on the line in the time slot. For a logic zero, a negative voltage is applied in the early part of the time slot followed by a positive voltage.

Figure 10: Manchester Biphase Encoding Principle



One advantage of Manchester coding is that the output power from the transmitter is constant regardless of the number of zeros or ones in the message. In the case of MIL STD 1553 transmission, a positive pulse is followed by a negative pulse or vice versa. This means that regardless of message content, the *average* voltage on the line is zero. This is important because a signal with zero *average* voltage can be coupled through a transformer, which is used in attaching the RT's to the data bus.

One characteristic of Manchester is helpful in receiving data. In the center of the time slot is a *transition* from positive or negative voltage. Another way of stating this; received voltage passes through zero in the center of each time slot. This can synchronize reception of data.

Since the system is asynchronous, there must be a method of identifying the beginning of a transmission. Recall in ARINC 429 there were at least four bit times of no activity on the bus, or idle time. This represents four or more wasted bit times. The *designator* for the beginning of a word in MIL STD 1553 is a pulse that *violates* Manchester *encoding*. A normal data bit is one polarity for one half the bit time and the opposite for the other half. The *state* of the data transmitted doesn't matter; there is always a change of *state* in the center of the time slot. In addition, the voltage is never positive or negative for more than the time of one bit.

Synchronizing is a Manchester pulse three times the data bit. No positive or negative pulse has a *width* of 1.5 bit time and the *wide* Manchester pulse is *recognized* as the synchronizing pulse. The pulse has an *average* voltage of 0, as do all other Manchester pulses and does not affect the zero *average* voltage.

The first data of a message is a command word transmitted by the bus controller. The remote terminal, RT, address immediately follows the synchronizing bit. Next is the T/R bit, which is set to a logic one if the RT is to transmit, and logic zero if it is to receive.

The RT address is a 5 bit binary number. Of the possible 32 *state*s of a 5 bit number, only 31 addresses are used. The number, 11111, is reserved for a broadcast mode. All RT's must receive messages destined for their own address and the broadcast address.

The 5 bits following the R/T bit are for a mode control indicator or subaddress for the RT. Of the possible 32 combinations of the subaddress field, two, 00000, 11111 are used for indicators that mode codes follow, leaving only 30 possible subaddresses. If the subaddress field does not *contain* a mode code indicator, the 5 bits following are a data word count. If the mode code is indicated, the 5 bits represent a mode. The final bit of the command word is a *parity* bit.

Following the command word are data words, in accordance with the data word count. The format for a data word is shown in Figure 11. The data word *contains* a start bit, 16 data bits and a *parity* bit.

Figure 11: MIL STD 1553 Data Word Format

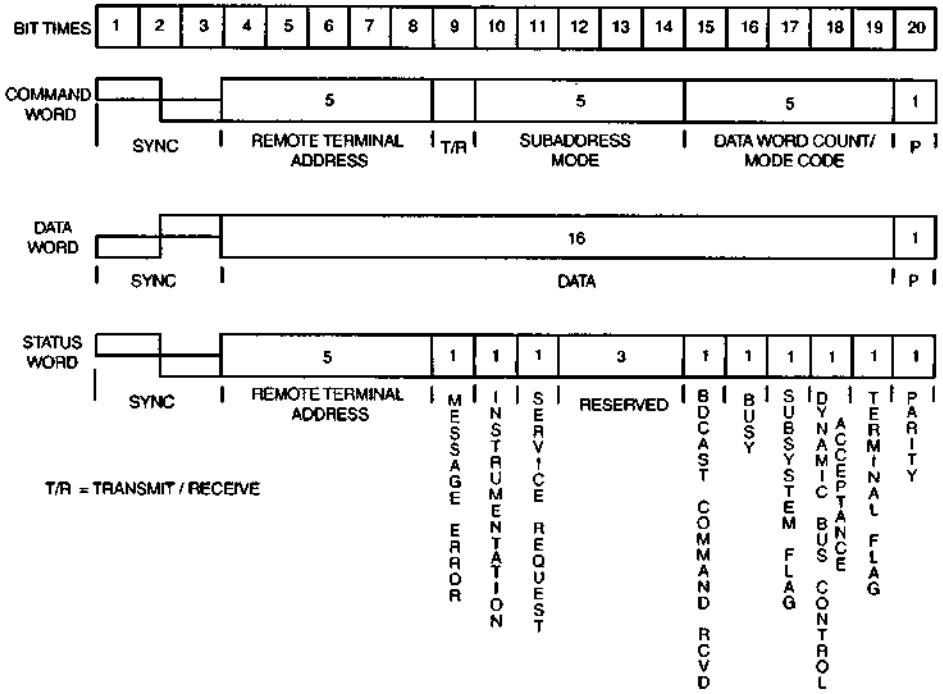


Figure 12: MIL STD 1553 RT Tap

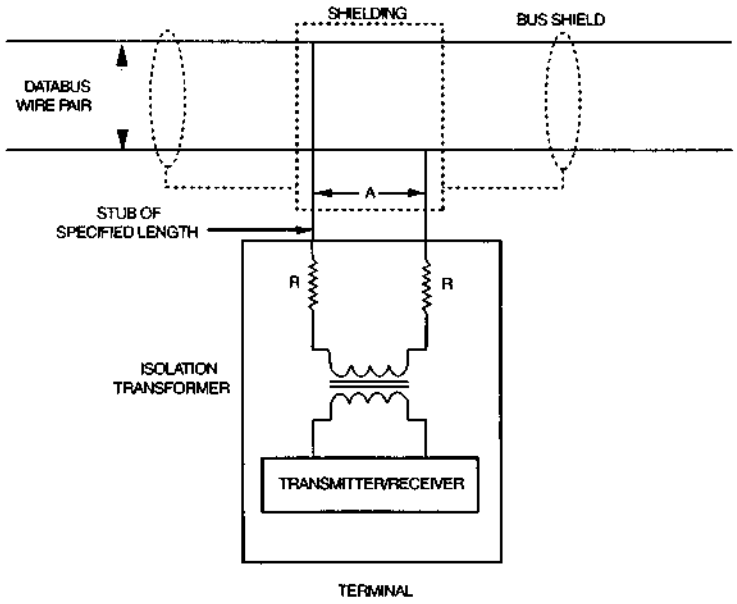


Figure 12 shows the connection of an RT to the 1533 bus with isolation resistors.

A transformer helps preserve the balance of the bus. As discussed, the ability of a balanced transmission line to reject *noise* from electromagnetic fields is a function of system balance.

The transformer connects directly to the bus through a *stub*. Even with the transformer, isolation resistors and *stubs* aid in isolating user equipment from the bus. Length of the *stub* is kept as short as practical, typically less than 5 meters.

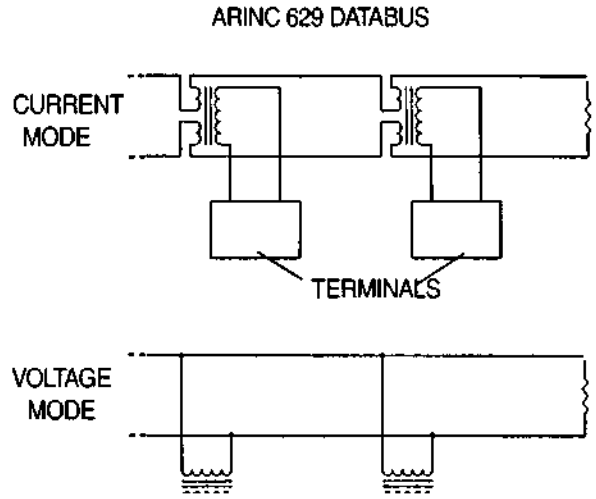
ARINC 629

A system similar to MIL STD 1553 but with significant *improvements* is ARINC 629. It operates at 2 Mb/s, twice the data rate of 1553. One attractive feature of ARINC 629 is that it will be defined for a fiber optic interface. ARINC 429 and MIL STD 1553 could be converted to light pulses in some fashion for *fiber*, but "some fashion" must be defined by an accepted specification.

The final significant difference between ARINC 629 and MIL STD 1553 is that ARINC 629 is defined for both voltage and current modes of operation. The systems discussed thus far use voltage coupling; receivers are high *impedance* and respond to voltage across the line. During a transmission, a voltage is placed on the bus. To prevent the bus from being loaded by RT's or receivers in the case of ARINC 429, there are series resistors between the terminal and bus for isolation. This is a problem because significant signal is lost in the resistors. An alternate to the voltage mode is the current mode.

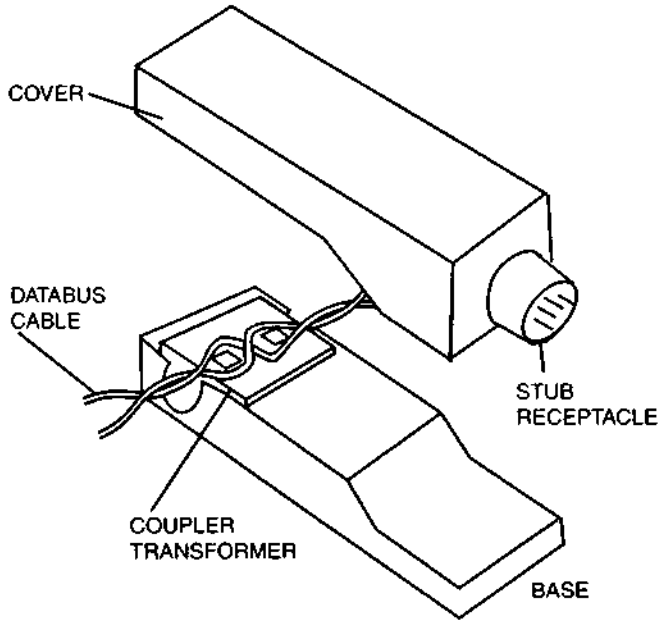
An example of current mode is shown in Figure 13 In this technique, the bus passes through the primary of a transformer. The *impedance* of the transformer is very low so there is no signal *loss*. In voltage mode transformer coupling, the transformer has a very high *impedance* so as not load down the bus. Because the transformer is in series with the bus in the current mode, a low *impedance* is desired.

Figure 13: Current and Voltage Coupling Modes for ARINC 629



What advantage is there to the current mode? The bus wire does not have to be broken to insert the RT and, if a unit is removed, the bus does not have to be reconnected. Figure 14 shows an ARINC 629 *coupler*. The bus passes through the transformer and becomes its primary. Only one half turn is applied to the primary, which implies very low *impedance*, exactly what is desired. To keep *loss* low, *impedance* in series with the bus should also be low.

Figure 14: ARINC 629 Bus Coupler

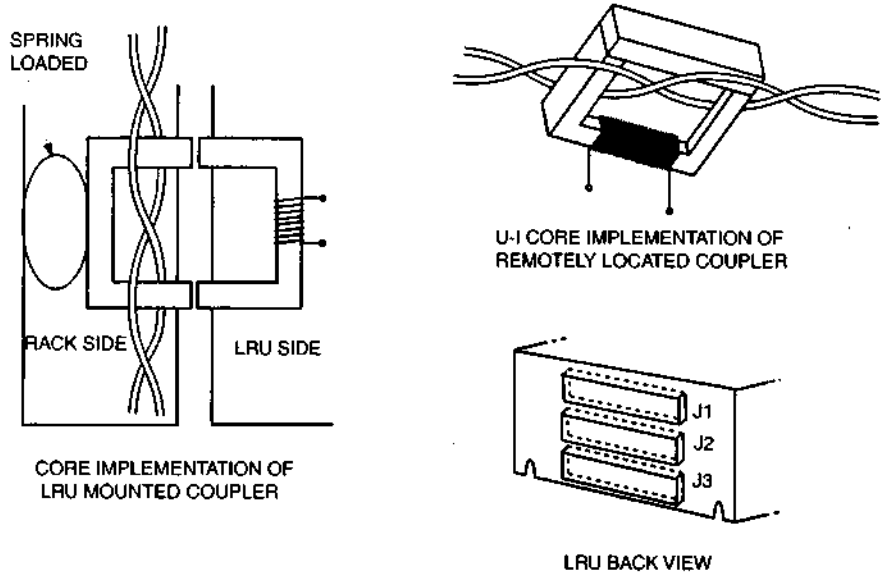


In addition to the *convenience* of adding and removing RT's without breaking the bus, the transformer *coupler* has the advantage of placing driver electronics in the coupling unit. This assures there is no *stub*, as found in ARINC 429 and MIL STD 1553 systems. Even though *stubs* in these systems are short, they are sources of signal *egress* and *ingress*. If eliminated, performance of the system is *enhanced*.

Another *coupler* type *provides convenience* for LRU's (line replaceable units). In this system, shown in Figure 15, the primary of the transformer is located in the LRU mount and the secondary is in the LRU itself. *Laminations* of the *coupler* are

separated so one half the transformer is located in the LRU with the second half in the mount. This couples the LRU to the bus magnetically without *disturbing* the bus. A unit on the bus can be removed and re-inserted while the bus is operating. Because the system continues to function, it's called a "hot *swap*". This technique would never *occur* during normal flight even in a military operation, but while troubleshooting on the ground.

Figure 15: Split Transformer ARINC 629 Coupler



Other Data Bus Standards

ARINC 629 is currently being used on Boeing 777 and will probably replace ARINC 429 as the most used communication standard in the future. Of course there are many other Data Bus Standards under ARINC as well as other standards used on aircrafts but not standardized by ARINC. Airbus uses both ARINC and other Data Bus Standards, specific for a certain type of equipment.

It is impossible to study all of these standards, for details on each type you can always refer to the ARINC Standard manuals or the manuals for another specific type of data bus system.

5.5 Logic Circuits

Representation of Binary Signal States

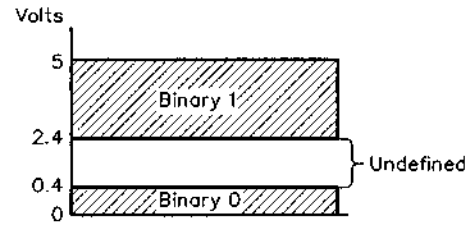
State Levels

Most digital systems use the binary number system because many simple physical systems are most easily described by two *state levels* (0 and 1). For example, the two *states* may represent "on" and "off," a *punched* hole or the absence of a hole in paper tape or a card, or a "mark" and "space" in a communications transmission. In electronic systems, *state levels* are physically represented by voltages. A typical choice is: *state* 0 = 0 V, *state* 1 = 5 V.

Since it is unrealistic to obtain these exact voltage values, a more practical choice is a range of values, such as: *state* 0 = 0.0 to 0.4 V, *state* 1 = 2.4 to 5.0 V.

Figure 1 illustrates this representation of *states* by voltage *levels*. The undefined region between the two binary *states* is also known as the *transition* region or *noise margin*.

Figure 1: Binary Signal States.



Representation of binary states 1 and 0 by a selected range of voltage levels

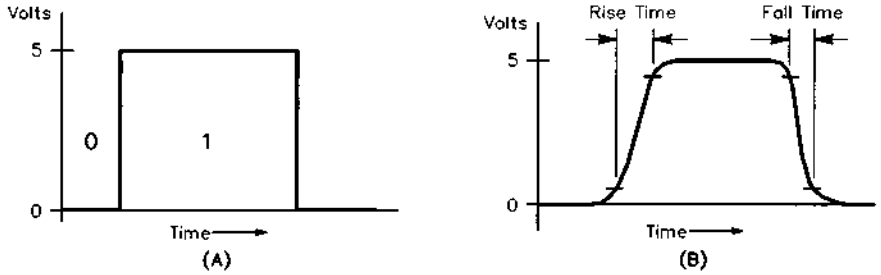
Transition Time

The *gap* in Figure 1 between binary 0 and binary 1, shows that a change in *state* does not *occur* instantly. There is a *transition* time between *states*. This *transition* time is a result of the time it takes to charge or discharge the *stray capacitance* in wires and other components because voltage cannot change instantaneously across a capacitor. (*Stray inductance* in the wires also has an effect because the current through an *inductor* can't change instantaneously.) The *transition* from a 0 to a 1 *state* is called the rise time. Similarly, the *transition* from a 1 to a 0 *state* is called the fall time. Note that these times need not be the same.

Figure 2 (A) shows an ideal signal, or pulse, with zero-time switching.

Figure 2 (B) shows a typical pulse, as it changes between *states* in a smooth curve. Rise and fall times vary with the logic family used and the location in a circuit. Typical values of *transition* time are in the microsecond to nanosecond range. In a circuit, distributed *inductances* and *capacitance* in wires or PC-board *traces* may cause rise and fall times to increase as the pulse moves away from the source.

Figure 2: Ideal and Actual Pulse.



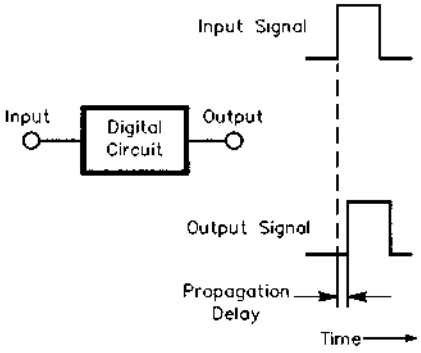
(A) An ideal digital pulse and (B) a typical actual pulse, showing the gradual *transition* **between** *states*

Propagation Delay

Rise and fall times only describe a *relationship* within a pulse. For a circuit, a pulse input into the circuit must *propagate* through the circuit; in other words it must pass through each component in the circuit until *eventually* it arrives at the circuit output. The time *delay* between providing an input to a circuit and to seeing a response at the output is the *propagation delay*, and is illustrated by Figure 3.

For modern switching logic, typical *propagation delay* values are in the 1 to 15 nanosecond range. (It is useful to remember that the *propagation delay* along a wire or printed-circuit-board *trace* is about 1.0 to 1.5 ns per inch.) *Propagation delay* is the result of cumulative *transition* times as well as transistor switching delays, reactive element charging times and the time for signals to travel through wires. In complex circuits, different *propagation* delays through different paths can cause problems when pulses must arrive somewhere at exactly the same time. Solutions to this timing problem include adding a *buffer* amplifier to the circuit and synchronization of circuits.

Figure 3: Propagation delay in a digital circuit.



Combinational Logic

Having defined a way to use voltage *levels* to physically represent digital numbers, we can apply digital signal theory to design useful circuits. Digital circuits combine binary inputs to produce a desired binary output or combination of outputs. This simple combination of 0's and 1's can become very powerful, *implementing* everything from simple switches to powerful computers.

A digital circuit falls into one of two types: combinational logic or sequential logic. In a combinational logic circuit, the output depends only on the present inputs. (If we ignore *propagation delay*.) In contrast, in a sequential logic circuit, the output depends on the present inputs, the *previous* sequence of inputs and often a clock signal.

The next section discusses combinational logic circuits. Later, we will build sequential logic circuits from the basics *established* here.

Boolean Algebra and the basic Logical Operators

Combinational circuits are composed of logic gates, which perform binary operations. Logic gates manipulate binary numbers, so you need an understanding of the algebra of binary numbers to understand how logic gates operate. Boolean algebra is the mathematical system to describe and design binary digital circuits. It is named after George Boole, the mathematician who developed the system. Standard algebra has a set of basic operations: addition, subtraction, multiplication and division. Similarly, Boolean algebra has a set of basic operations, called logical operations: NOT, AND and OR.

The function of these operators can be described by either a Boolean equation or a truth table.

A Boolean equation describes an operator's function by representing the inputs and the operations performed on them. An equation is of the form "B = A," while an expression is of the form "A." In an *assignment* equation, the inputs and operations appear on the right and the result, or output, is *assigned* to the variable on the left.

A truth table describes an operator's function by listing all possible inputs and the *corresponding* outputs. Truth tables are sometimes written with T's and F's (for true and false) or with their respective *equivalents*, 1s and 0s. In company data books (catalogues of logic devices a company manufactures), truth tables are usually written with H's and L's (for high and low). In the figures, 1 will mean high and 0 will mean low. This representation is called positive logic. The meaning of different logic types and why they are useful is discussed in a later section.

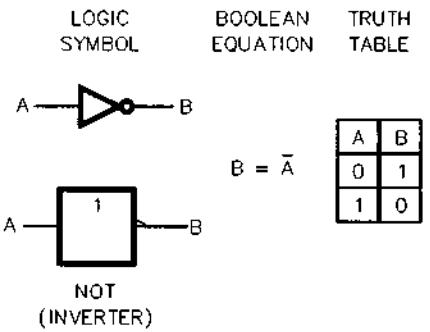
Each Boolean operator also has two circuit symbols *associated* with it. The traditional symbol appears on top in each of the figures; for example, the triangle and bubble for the NOT function in Figure 4. In the traditional symbols, a small circle, or bubble, always represents "NOT." (This bubble is called a *state* indicator.) Appearing just below the traditional symbol is the newer *ANSI, IEEE* Standard symbol. This symbol is always a square box with notations inside it. In these newer symbols, a small flag represents "NOT." The new notation is an attempt to replace the detailed logic drawing of a complex function with a simpler block symbol.

Figure 4, Figure 5 and Figure 6 show the truth tables, Boolean algebra equations and circuit symbols for the three basic Boolean operations: NOT, AND and OR.

All combinational logic functions, no matter how complex, can be described in terms of these three operators.

The NOT operation is also called *inversion, negation* or *complement*. The circuit that *implements* this function is called an inverter or inverting *buffer*. The most common notation for NOT is a bar over a variable or expression. For example, NOT A is denoted $\bar{A}$. This is read as either "Not A" or as "A bar." A less common notation is to denote Not A by A', which is read as "A prime."

Figure 4: Inverter.

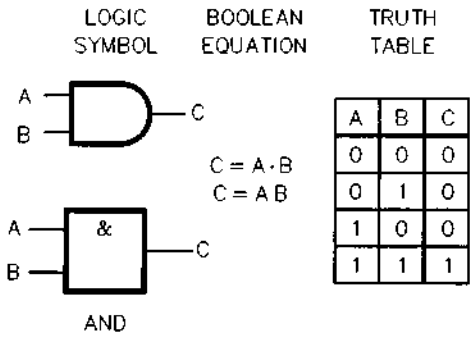


A	B
0	1
1	0

While the inverting *buffer* and the non inverting *buffer* covered later have only one input and output, many combinational logic elements can have multiple inputs. When a combinational logic element has two or more inputs and one output, it is called a **gate**. For simplicity, the figures and truth tables for multiple-input elements will show the operations for only two inputs, the minimum number.

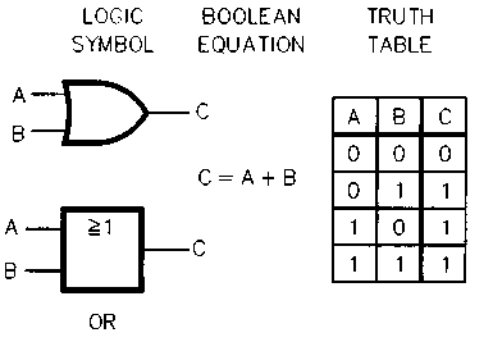
The output of an **AND function** is **1** only if all of the inputs are 1. Therefore, if any of the inputs are 0, then the output is 0. The notation for an AND is either a *dot* (•) between the inputs, as in C = A•B, or nothing between the inputs, as in C = AB. Read these equations as "C equals A AND B."

Figure 5: Two-input AND gate.



The OR gate detects if one or more inputs are 1. In other words, if any of the inputs are 1, then the output of the OR gate is 1. Since this includes the case where more than one input may be 1, the OR operation is also known as an INCLUSIVE OR. The OR operation detects if at least one input is 1. Only if all the inputs are 0, then the output is 0. The notation for an OR is a plus sign (+) between the inputs, as in C = A + B. Read this equation as "C equals A OR B."

Figure 6: Two-input OR gate.

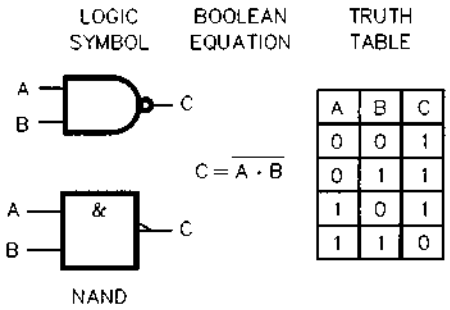


Other Common Gates

More complex logical functions are *derived* from combinations of the basic logical operators. These operations — NAND, NOR, XOR, XNOR and the non inverter — are illustrated in Figure 7 through Figure 11 respectively. As before, each is described by a truth table, Boolean algebra equation and circuit symbols. Also as before, except for the non inverter, each could have more inputs than the two illustrated.

The NAND gate (short for NOT AND) is *equivalent* to an AND gate followed by a NOT gate. Thus, its output is the *complement* of the AND output: The output is a 0 only if all the inputs are 1. If any of the inputs is 0, then the output is a 1.

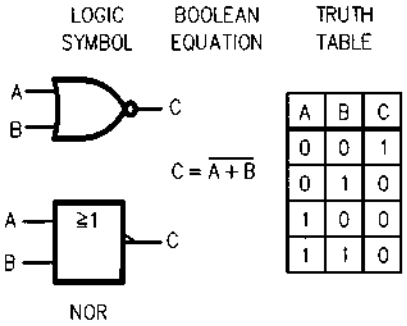
Figure 7: Two-input NAND gate.



A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

The NOR gate (short for NOT OR) is *equivalent* to an OR gate followed by a NOT gate. Thus, its output is the *complement* of the OR output: If any of the inputs are 1, then the output is a 0. Only if all the inputs are 0, then the output is a 1.

Figure 8: Two-input NOR gate.



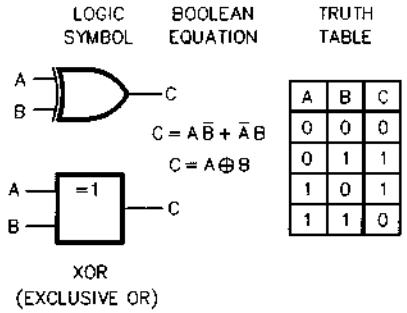
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

The operations so far enable a designer to *determine* two general cases: (1) if all inputs have a desired *state* or (2) if at least one input has a desired *state*. The XOR and XNOR gates enable a designer to *determine* if one and only one input of a desired *state* is present.

The XOR gate (read as EXCLUSIVE OR) has an output of 1 if one and only one of the inputs is a 1 *state*. The output is 0 otherwise. The symbol for XOR is ⊕. This is easy to remember if you think of the "+" OR symbol enclosed in an "O" for only one.

The XOR gate is also known as a "half adder," because in binary arithmetic it does everything but the "*carry*" operation. The Truth Table in Figure 9 show the possible binary additions for a two-input XOR.

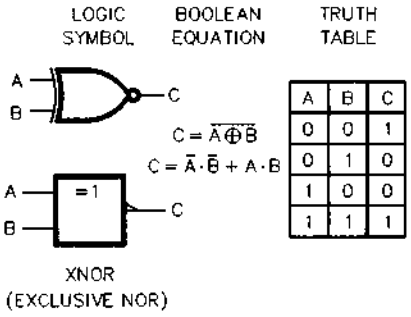
Figure 9: Two-input XOR gate.



A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

The XNOR gate (read as EXCLUSIVE NOR) is the *complement* of the XOR gate. The output is 0 if one and only one of the inputs is a 1. The output is 1 either if all inputs are 0 or more than one input is 1.

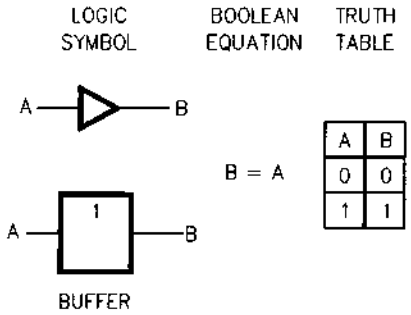
Figure 10: Two-input XNOR gate.



A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

A non inverter, also known as a *buffer*, amplifier or driver, at first *glance* does not seem to do anything. It simply receives an input and produces the same output. In reality, it is changing other properties of the signal in a useful fashion, such as amplifying the current *level*. The practical uses of a non inverter include (A) providing sufficient current to drive a number of gates, (B) interfacing between two logic families, (C) obtaining a desired pulse rise time and (D) providing a slight *delay* to make pulses arrive at the proper time.

Figure 11: Nonirritating buffer.



A	B
0	0
1	1

Boolean Algebra Laws and Rules

The analysis of a circuit starts with a logic diagram and then *derives* a circuit description. In digital circuits, this description is in the form of a truth table or logical equation. The synthesis, or design, of a circuit goes in the reverse: starting with an informal description, determining an equation or truth table and then expanding the truth table to components that will *implement* the desired response.

In both of these *processes*, we need to either simplify or expand a complex logical equation.To manipulate an equation, we use mathematical theorems. Theorems are statements that have been proven to be true. The theorems of Boolean algebra are very similar to those of standard algebra, such as commutativity and associativity. Proofs of the Boolean algebra theorems can be found in an introductory digital design textbook.

Basic Theorems

Figure 12 lists the theorems for a single variable and Figure 13 lists the theorems for two or more variables. These tables illustrate the principle of duality exhibited by the Boolean theorems: Each theorem has a dual in which, after *swapping* all AND's with OR's and all 1s with 0s, the statement is still true.

Figure 12: Single Variable Theorems

Identities:	$A \cdot 1 = A$	$A + 0 = A$
Null elements:	$A \cdot 0 = 0$	$A + 1 = 1$
Idempotence:	$A \cdot A = A$	$A + A = A$
Complements:	$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$
Involution:	$\overline{(\overline{A})} = A$	

The tables also illustrate the *precedence* of the Boolean operations: the order in which operations are performed when not specified by parenthesis. From highest to lowest, the *precedence* is NOT, AND then OR. For example, the distributive law includes the expression "A + B•C." This is *equivalent* to "A + (B•C)." The parenthesis around (B•C) can be left out since an AND operation has higher priority than an OR operation. *Precedence* for Boolean algebra is similar to the *convention* of standard algebra: *raising to a power*, then multiplication, then addition.

Figure 13: Multivariable Theorems

Commutativity:	$A \cdot B = B \cdot A$ $A + B = B + A$
Associativity:	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$ $(A + B) + C = A + (B + C)$
Distributivity:	$(A + B) \cdot (A + C) = A + B \cdot C$ $A \cdot B + A \cdot C = A \cdot (B + C)$
Covering:	$A \cdot (A + B) = A$ $A + A \cdot B = A$
Combining:	$(A + B) \cdot (A + \overline{B}) = A$ $A \cdot B + A \cdot \overline{B} = A$
Consensus:	$A \cdot B + \overline{A} \cdot C + B \cdot C = A \cdot B + \overline{A} \cdot C$ $(A + B) \cdot (\overline{A} + C) \cdot (B + C) = (A + B) \cdot (\overline{A} + C)$

DeMorgan's Theorem

One of the most useful theorems in Boolean algebra is DeMorgan's Theorem:
$\overline{A \bullet B} = \overline{A} + \overline{B}$ and its dual: $\overline{A + B} = \overline{A} \bullet \overline{B}$.

The truth table in Figure 14 proves these statements.

Figure 14: DeMorgan's Theorem

DeMorgan's Theorem

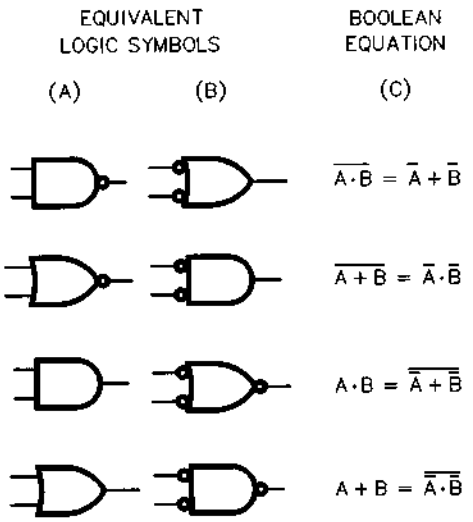
(A) $\overline{A \bullet B} = \overline{A} + \overline{B}$
(B) $\overline{A + B} = \overline{A} \bullet \overline{B}$
(C)

(1) A	(2) B	(3) $\overline{A}$	(4) $\overline{B}$	(5) A • B	(6) $\overline{A \bullet B}$	(7) A + B	(8) $\overline{A+B}$	(9) $\overline{A} \bullet \overline{B}$	(10) $\overline{A}+\overline{B}$
0	0	1	1	0	1	0	1	1	1
0	1	1	0	0	1	1	0	0	1
1	0	0	1	0	1	1	0	0	1
1	1	0	0	1	0	1	0	0	0

(A) and (B) are statements of DeMorgan's Theorem. The truth table at (C) is proof of these statements: (A) is proven by the equivalence of columns 6 and 10 and (B) by columns 8 and 9.

DeMorgan's Theorem *provides* a way to simplify the *complement* of a large expression. It also enables a designer to interchange a number of *equivalent* gates, as shown by Figure 15.

Figure 15: Interchanging of equivalent gates.



Each gate in column A is *equivalent* to the opposite gate in column B. The Boolean equations in column C formally *state* the *equivalencies*

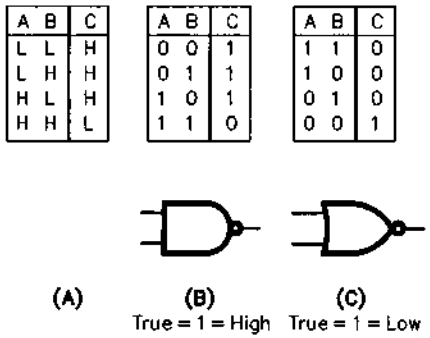
The *equivalent* gates show that the duality principle works with symbols the same as it does for Boolean equations: just *swap* AND's with OR's and switch the bubbles. For example, the NAND gate — an AND gate followed by an inverter bubble — becomes an OR gate preceded by two inverter bubbles. DeMorgan's Theorem is important because it means any logical function can be *implemented* using either inverters and AND gates or inverters and OR gates. Also, the ability to change placement of the bubbles using DeMorgan's theorem is useful in dealing with mixed logic, to be discussed next.

Positive and Negative Logic

The truth tables shown in the figures in this chapter are drawn for positive logic. In positive logic, or high true, a higher voltage means true (logic 1) while a lower voltage means false (logic 0). This is also referred to as active high: a signal performs a named action or denotes a condition when it is "high" or 1.

In negative logic, or low true, a lower voltage means true (1) and a higher voltage means false (0). An active low signal performs an action or denotes a condition when it is "low" or 0. In both logic types, true = 1 and false = 0; but whether true means high or low differs. Company data books are drawn for general truth tables: an "H" for high and an "L" for low. (Some tables also have an "X" for a "don't care" *state*.) The function of the table can differ depending on whether it is interpreted for positive logic or negative logic. Figure 16 shows how a general truth table differs when interpreted for different logic types. The same truth table gives two *equivalent* gates: positive logic gives the function of a NAND gate while negative logic gives the function of a NOR gate.

Figure 16: Positive and Negative Logic



(A) A general truth table, (B) a truth table and NAND symbol for positive logic and (C) a truth table and NOR symbol for negative logic

Note that these gates *correspond* to the *equivalent* gates from DeMorgan's theorem. A bubble on an input or output terminal indicates an active low device. The absence of bubbles indicates an active high device.

Like the bubbles, signal names can be used to indicate logic *states*. These names can aid the understanding of a circuit by indicating control of an action (GO, /ENABLE) or *detection* of a condition (READY, /ERROR).

The action or condition occurs when the signal is in its active *state*. When a signal is in its active *state*, it is called *asserted*; a signal not in its active *state* is called negated or de-*asserted*. A prefix can easily indicate a signal's active *state*: active low signals are preceded by a "/," like /READY, while active high signals have no prefix. Standard practice is that the signal name and input pin match (have the same active *level*). For example, an input with a bubble (active low) may be called /READY while an input with no bubble (active high) is called READY. Output signal names should always match the device output pin.

In this chapter, positive logic is used unless indicated otherwise. Although using mixed logic can be confusing, it does have some advantages. Mixed logic combined with DeMorgan's Theorem can *promote* more effective use of available gates. Also, well-chosen signal names and placement of bubbles can *promote* more understandable logic diagrams.

Sequential / Combinational Logic

The *previous* section discussed combinational logic, whose outputs depend only on the present inputs. In contrast, in sequential logic circuits, the new output depends not only on the present inputs but also on the present outputs. The present outputs depended on the *previous* inputs and outputs and those earlier outputs depended on even earlier inputs and outputs and so on. Thus, the present outputs depend on the *previous* sequence of inputs and the system has memory. Having the outputs become part of the new inputs is known as feedback.

This section first introduces a number of terms necessary to understand sequential logic: types of synchronicity, types of control signals and ways to illustrate circuit function. Numerous sequential logic circuits are then introduced. These circuits *provide* an overview of the basic sequential circuits that are commercially available.

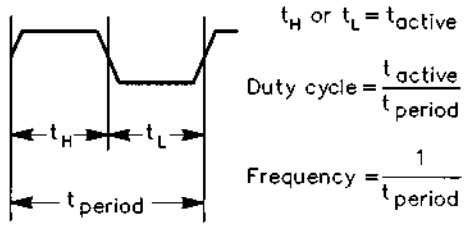
Synchronicity and Control Signals

When a combinational circuit is given a set of inputs, the outputs take on the expected values after a *propagation delay* during which the inputs travel through the circuit to the output. In a sequential circuit, however, the travel through the circuit is more complicated. After application of the first inputs and one *propagation delay*, the outputs take on the resulting *state*; but then the outputs start trickling back through and, after a second *propagation delay*, new outputs appear. The same happens after a third *propagation delay*. With *propagation* delays in the nanosecond range, this cycle around the circuit is rapidly and continually generating new outputs. A user needs to know when the outputs are valid.

There are two types of sequential circuits: **synchronous** circuits and **asynchronous** circuits, which are analysed differently for valid outputs. In asynchronous operation, the outputs respond to the inputs immediately after the *propagation delay*. To work properly, this type of circuit must *eventually* reach a stable *state*: the inputs and the fed back outputs result in the new outputs staying the same. When the non feedback inputs are changed, the feedback cycle needs to *eventually* reach a new stable *state*.

In synchronous operation, the outputs change *state* only at specific times. These times are *determined* by the presence of a particular input signal: a clock, toggle, latch or enable. Synchronicity is important because it ensures proper timing: all the inputs are present where needed when the control signal causes a change of *state*.

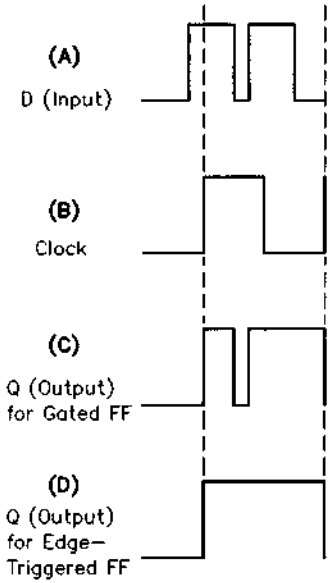
Figure 17: Clock signal terms.



The duty cycle would be t_h / t_{period} for an active high signal and t_l / t_{period} for an active low signal.

The clock period is the time between *successive transitions* in the same direction; the clock frequency is the reciprocal of the period. A pulse or clock tick is the first *edge* in a clock period, or sometimes the period itself or the first half of the period. The duty cycle is the percentage of time that the clock signal is at its *asserted level*. The reaction of a synchronous circuit to its control signal is static or dynamic. Static, gated or *level-triggered* control allows the circuit to change *state* whenever the control signal is at its active or *asserted level*. Dynamic, or *edge-triggered*, control allows the circuit to change *state* only when the control signal changes from un *asserted* to *asserted*. By *convention*, a control signal is active high if *state* changes *occur* when the signal is high or at the rising *edge* and active low in the opposite case. Thus, for positive logic, the *convention* is enable = 1 or enable goes from 0 to 1. This *transition* from 0 to 1 is called positive *edge-triggered* and is indicated by a small triangle inside the circuit box. A circuit responding to the opposite *transition*, from 1 to 0, is called negative *edge-triggered*, indicated by a bubble with the triangle. Whether a circuit is *level-triggered* or *edge-triggered* can affect its output, as shown by Figure 18. Input D includes a very brief pulse, called a *glitch*, which may be caused by *noise*. The differing results at the output illustrate how *noise* can cause errors.

Figure 18: Level-triggered vs edge-triggered for a D flip-flop:



(A) input D, (B) clock input, (C) output Q for *level-triggered*: circuit responds whenever clock is 1. (D) output Q for *edge-triggered*: circuit responds only at rising *edge* of clock. Notice that the short negative pulse on the input D is not reproduced by the *edge-triggered flip-flop*.

Edge-Triggered Flip-Flop

The *edge-triggered flip-flop* solves the problem of *noise*. An example of *noise* is the *glitch* shown in Figure 18. The different outputs for the *level* and *edge-triggered* methods in this figure show how a *glitch* can cause an output error. *Edge-triggering* avoids the problem of *noise* by minimizing the time during which a circuit responds to its inputs: the chance of a *glitch* occurring during the nanosecond *transition* of a clock pulse is remote. A side benefit of *edge-triggering* is that only one new output is produced per clock period. *Edge-triggering* is denoted by a small rising-*edge* or falling-*edge* symbol such as ⌡ or ⌝ ; sometimes an arrow is included such as ↑ or ↓

This symbol appears in the circuit's truth table and can also appear, *instead* of the clock triangle, inside the schematic symbol.

Flip-Flops

Flip-flops are the basic building blocks of sequential circuits. A *flip-flop* is a device with two stable *states*: the set *state* (1) or the reset *state* (0). (The reset *state* is also called the cleared *state*.) The *flip-flop* can be placed in one or the other of the two *states* by *applying* the *appropriate* input. (Since a common use of *flip-flops* is to store one bit of information, some use the term **latch** interchangeably with *flip-flop*. A set of latches, or *flip-flops* holding an n-bit number is called a register.) While gates have special symbols, the schematic symbol for most components is a rectangular box with the circuit name or abbreviation, the signal names and assertion bubbles. For *flip-flops*, the circuit name is usually omitted since the signal names are enough to indicate a *flip-flop* and its type. The four basic types of *flip-flops* are the S-R, D, T and J-K. The first section examines the S-R *flip-flop* for each of the various control methods. The next section introduces each of the other basic *flip-flops* and their uses.

S-R Flip-Flop

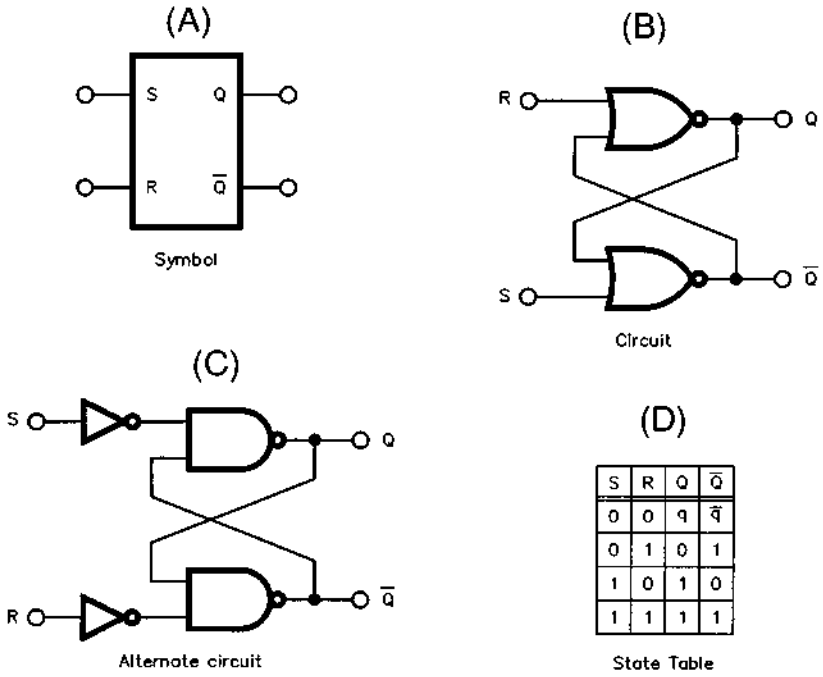
The S-R *flip-flop* is one of the simplest circuits for storing a bit of information. It has two inputs, represented by S (set) and R (reset). These inputs, naturally, cause the two possible output *states*: if S = 1 and R = 0, then output Q is set to 1; if S = 0 and R = 1, then output Q is reset to 0; if both inputs are 0, then the output remains unchanged; and if both inputs are 1, then the output cannot be *determined*.

The S-R *flip-flop* can illustrate each of the types of control signals: unclocked (asynchronous, no control signal), clocked or gated, master-slave and *edge-triggered*.

Unclocked/Sequential

The unclocked S-R *flip-flop*, shown in Figure 19, is an asynchronous device; its outputs change immediately to reflect changes on its inputs. The circuit *consists* of two NOR gates. The sequential nature of the circuit is a result of the output of each NOR gate being fed back as an input to the opposite gate. The *state transition* table shows the expected set/reset pattern of inputs to outputs. The table shows an *unpredictable* result for inputs S = 1 and R = 1. In actual circuits, the results vary and are usually either Q = $\overline{Q}$ = 1 or Q = $\overline{Q}$ = 0. While Q = $\overline{Q}$ is a logical impossibility, real *flip-flops* may present this output. The designer should avoid the R = S = 1 input and make no *assumptions* about the resulting output. The *flip-flop* is not *predictable* if both inputs go to 0 at exactly the same time. Figure 19C shows an alternate *implementation* of the S-R *flip-flop*, with two NAND gates and two inverters. Since a NAND gate can become an inverter by having its two inputs receive the same signal, the S-R *flip-flop* can be *implemented* with four NAND gates. This alternate version is important because a 4-NAND gate chip is one of the most readily available commercial integrated circuits; thus, the 4-NAND gate S-R *flip-flop* can be *implemented* on a single *IC*.

Figure 19: Unclocked S-R Flip-Flop.

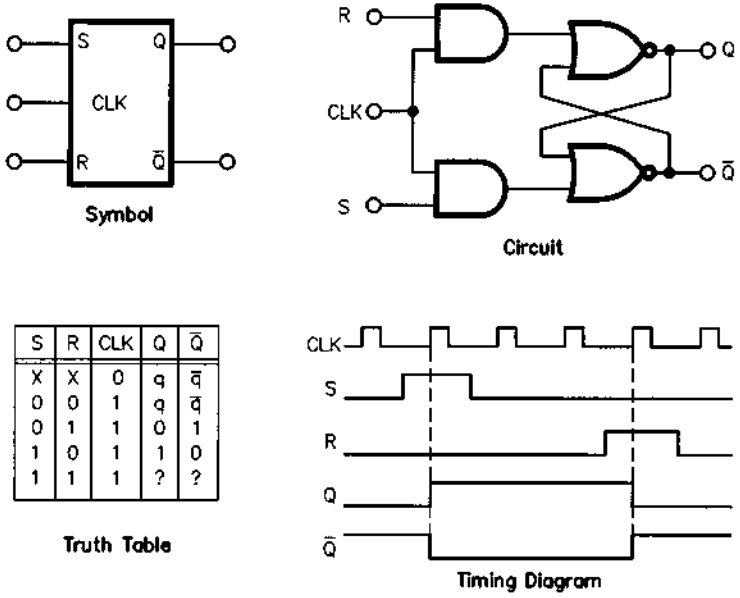


Gated or Level-Triggered

The gated S-R *flip-flop*, or gated latch, has a controlling input in addition to its S and R inputs. The inputs S and R produce the same results as those on an unclocked S-R *flip-flop*, but a change in output will only *occur* when the control input is high. A gated S-R *flip-flop* is illustrated in Figure 20 along with a timing diagram for a clock input. This *flip-flop* is also called the R-S-T *flip-flop*, where "T," for toggle, is the clock input. Although not often used, the R-S-T *flip-flop* is important because it illustrates a step between the R-S *flip-flop* and the J-K *flip-flop*. A problem with the *level-triggered flip-flop* is that the Q output can change more than once while the clock is *asserted*.

We would prefer the output to change only once per clock period for easier timing design. A second problem can *occur* when *flip-flop*s are connected in series and *triggered* by the same clock pulse or, similarly, when a *flip-flop* is in series with itself, using its own output as an input. Since the series-connected *flip-flop* feeds back to itself, its output will be changing at about the same time as it receives new input. This can result in an erroneous output.

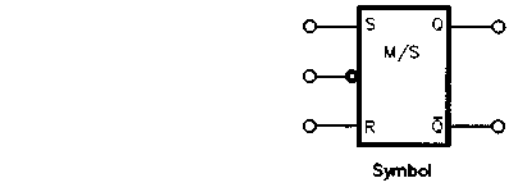
Figure 20: Clocked S-R *Flip-Flop*.



Master/Slave Flip-Flop

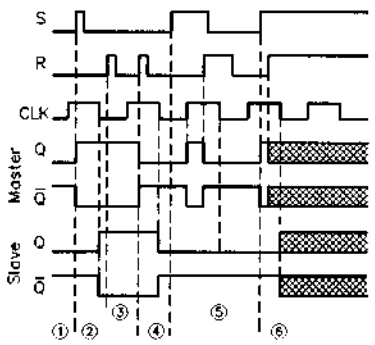
A solution to the problems of the *level-triggered* method is a circuit that samples and stores its inputs before changing its outputs. Such a circuit is built by placing two *flip-flops* in series; both *flip-flops* are *triggered* by a common clock but an inverter on the second *flip-flop*'s clock causes it to be *asserted* only when the first *flip-flop* is not *asserted*. The action for a given clock pulse is as follows: The first, or master, *flip-flop* is active when the clock is high, sampling and storing the inputs. The second, or slave, *flip-flop* gets its input from the master and acts when the clock is low. *Hence*, when the clock is 1, the input is sampled; then when the clock becomes 0, the output is generated. A master/slave *flip-flop* is built with either two S-R *flip-flops*, as shown by Figure 21, or with two J-K *flip-flops*. Note that a bubble appears on the schematic symbol's clock input, reminding us that the output appears when the clock is *asserted* low. This is *conventional* for TTL-style J-K *flip-flops*, but it can be different for CMOS devices. The master/slave method isolates output changes from input changes, eliminating the problem of series-fed circuits. It also ensures only one new output per clock period, since the slave *flip-flop* responds to only the single sampled input. A problem can still *occur*, however, because the master *flip-flop* can change more than once while it is *asserted*; thus, there is the potential for the master to sample at the wrong time. There is also the potential that either *flip-flop* can be affected by *noise*.

Figure 21: Master/Slave S-R *Flip-Flop*.

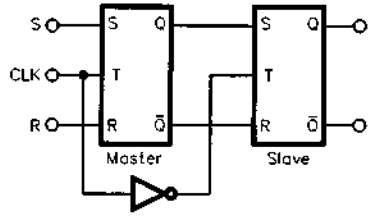


Symbol

Timing Diagram



① Assumed last states.
② Master is set to 1. Slave follows on Clocks Falling Edge.
③ No Change Since Clock is 0
④ Master is reset to 0. Slave follows on clocks falling edge.
⑤ Note how master only responds while clock is high. Slave only responds when clock gets low.
⑥ Unknown state since both R and S are 1.



M/S Flip-Flop Made of R-S Flip-Flops and an Inverter

Circuit

S	R	Master Qn	Slave Qn	Master Qn+1	Slave Qn+1
X	X	Q	Q	Q	Q
0	0	Q	Q	Q	Q
1	0	1	Q	1	1
0	1	0	Q	0	0
1	1	?	?	?	?
Clock:					

Qn : Output when clock is high
Qn+1 : Output when clock gets low

Truth Table

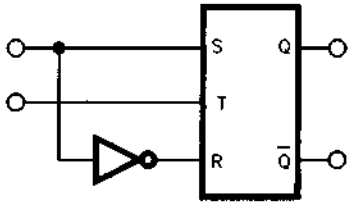
Summary of Standard Flip-flops

Figure 23 *provides* a summary of the four basic *flip-flops*: the S-R (Set-Reset), D (Data or *Delay*), T (Toggle) and J-K. Each is briefly explained below, including its particular applications. The internal circuitry of each of these *flip-flops* is similar to the components and complexity of the S-R *flip-flop*.

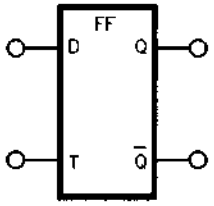
D Flip-Flop

In a D (data) *flip-flop*, the data input is transferred to the outputs when the *flip-flop* is enabled: The logic *level* at input D is transferred to Q when the clock is positive; the Q output retains this logic *level* until the next positive clock pulse (see Figure 22). The *flip-flop* is also called a *delay flip-flop* because, once enabled, it passes D after a *propagation delay*. A D *flip-flop* is useful to store one bit of information. A collection of D *flip-flops* forms a register.

Figure 22: D Flip-Flop



R–S–T Flip–Flop and Inverter



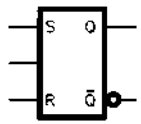
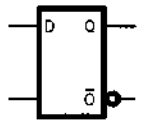
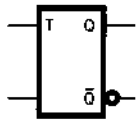
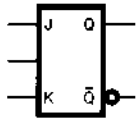
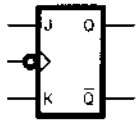
Prewired D Flip–flop

T	D	Q	Q̄
0	0	q	q̄
0	1	q	q̄
1	0	0	1
1	1	1	0

Truth Table

Figure 23: Summary of Standard Flip-flops

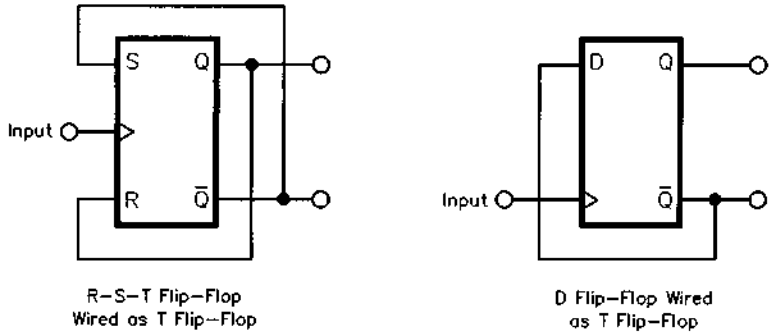
q = current state Q = next state X = don't care

Flip–Flop Type	Symbol	Truth Table	Characteristic Equation	Excitation Table
SR		S R CLK | Q X X 0 | q 0 0 1 | q 0 1 1 | 0 1 0 1 | 1 1 1 1 | ?	$Q = \left(S + \bar{R} \cdot q\right) \cdot CLK$ with $S \cdot R = 0$	q Q | S R CLK 0 0 | 0 X 1 0 1 | 1 0 1 1 0 | 0 1 1 1 1 | X 0 1
D		D CLK | Q X 0 | q 0 1 | 0 1 1 | 1	$Q = D \cdot CLK$	q Q | D CLK 0 0 | X 0 0 1 | 1 1 1 0 | 0 1 1 1 | X 0
T		T CLK | Q X 0 | q 0 1 | q 1 1 | q̄	$Q = (T \oplus q) \cdot CLK$	q Q | T CLK 0 0 | 0 X 0 1 | 1 1 1 0 | 1 1 1 1 | 0 X
JK		J K CLK | Q X X 0 | q 0 0 1 | q 0 1 1 | 0 1 0 1 | 1 1 1 1 | q̄	$Q = \left(J \cdot \bar{q} + \bar{K} \cdot q\right) \cdot CLK$	q Q | J K CLK 0 0 | 0 X 1 0 1 | 1 X 1 1 0 | X 1 1 1 1 | X 0 1
JK		J K CLK | Q 0 0 ⊓ | 0 0 1 ⊓ | 0 1 0 ⊓ | 1 1 1 ⊓ | q̄	$Q = \left(J \cdot \bar{q} + \bar{K} \cdot q\right) \cdot CLK$	q Q | J K CLK 0 0 | 0 X ⊓ 0 1 | 1 X ⊓ 1 0 | X 1 ⊓ 1 1 | X 0 ⊓

Toggle Flip-Flop

In a T *flip-flop*, the output toggles (changes *state*) with each positive clock pulse. The T *flip-flop* is also called a *complementing flip-flop*. Fig 7.24 shows how a T *flip-flop* can be created from either an S-R or D *flip-flop*. The timing diagram in Fig 7.24 shows an important result of the T-*flip-flop*: the output frequency is one half of the input frequency. Thus, a T *flip-flop* is a 2:1 (also called modulo-2 or radix-2) frequency divider. Two T *flip-flops* connected in series form a 4:1 divider and so on.

Figure 24: Toggle Flip-Flop



J-K Flip-Flop

It's somewhat ironic that the most readily available *flip-flop*, the J-K *flip-flop*, is discussed last and so briefly. The discussion is short because the J-K *flip-flop* acts the same as the S-R *flip-flop* (where J = S and K = R) with only one difference: The S-R *flip-flop* had the disadvantage of invalid results for the inputs 1,1. For the J-K *flip-flop*, simultaneous 1,1 inputs cause Q to change *state* after the clock *transition*.

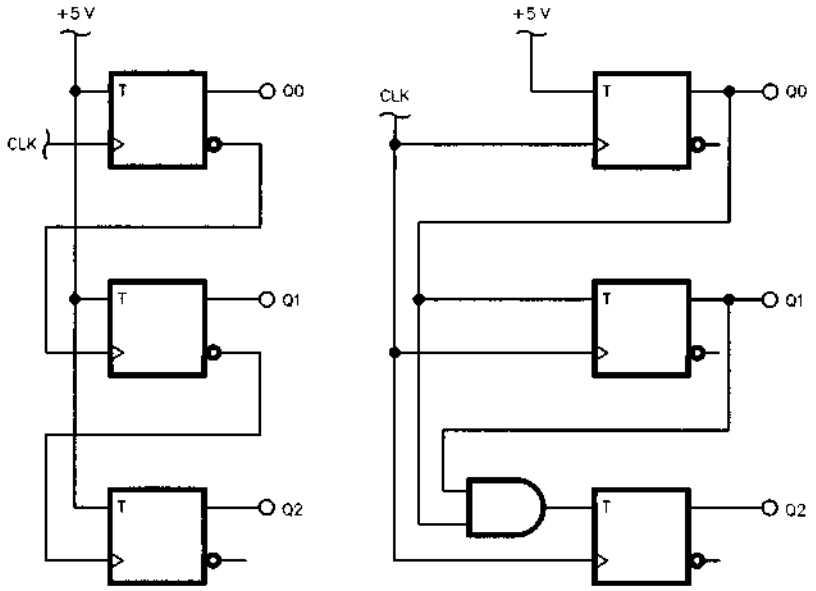
Summary

Only the D and J-K *flip-flops* are generally available as commercial *integrated circuit* chips. Since memory and temporary storage are so often desirable, the D *flip-flop* is manufactured as the simplest way to *provide* memory. When more functionality is needed, the J-K *flip-flop* is available. The J-K *flip-flop* can substitute for an S-R *flip-flop* and a T *flip-flop* can be created from either the D or J-K *flip-flop*.

Counters

Groups of *flip-flops* can be combined to make counters. Toggle *flip-flops* are the most common for *implementing* a counter. Intuitively, a counter is a circuit that starts at *state* 0 and sequences up through *states* 1, 2, 3, to m, where m is the maximum number of *states* available. From *state* m, the next *state* will return the counter to 0. This describes the most common counter: the n-bit binary counter, with N outputs *corresponding* to 2^N = m *states*. Such a counter can be made from N *flip-flops*, as shown in Fig 7.25. This figure shows *implementations* for each of the types of synchronicity. Both circuits pass the data count from stage to stage. In the asynchronous counter, Figure 25(A) the clock is also passed from stage to stage and the circuit is called ripple or ripple-*carry*. In the synchronous counter, Figure 25(B), each stage is controlled by a common clock signal.

Figure 25: Three-bit binary counter



Asynchronous "Ripple" Counter

Synchronous Counter

(The output of each flip-flop: toggles whenever T is high and clock is high)
stays the same whenever T is low

There are numerous variations on this first example of a counter. Most counters have the ability to clear the count to 0. Some counters can also preset to a desired count. The clear and preset control inputs are often asynchronous — they change the output *state* without being clocked. Counters may either count up (*increment*) or down (*decrement*). Up/down counters can be controlled to count in either direction. Counters can have sequences other than the standard numbers, for example a BCD counter. Counters are also not restricted to changing *state* on every clock cycle. An n-bit counter that changes *state* only after m clock pulses is called a divider or divide-by-m counter. There are still 2^N = m *states*; however, the output after p clock pulses is now p / m. Combining different divide-by-m counters can result in almost any desired count. For example, a *base* 12 counter can be made from a divide-by-2 and a divide-by-6 counter; a *base* 10 (decade) counter *consists* of a divide-by-2 and a BCD divide-by-5 counter.

The outputs of these counters are binary. To produce output in decimal form, the output of a counter would be *provided* to a binary-to-decimal decoder chip and/or an LED display.

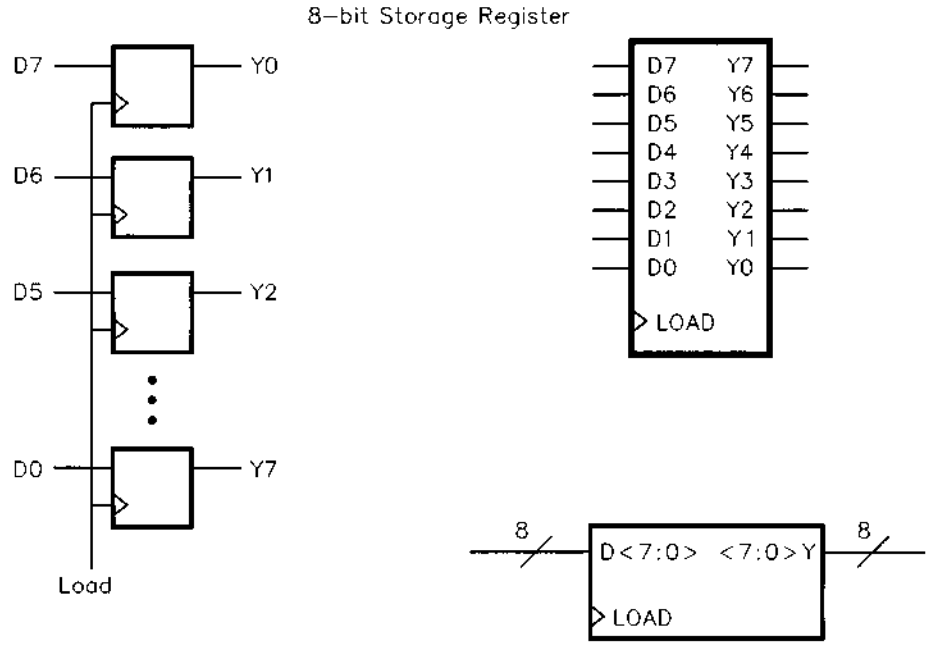
Registers

Groups of *flip-flops* can be combined to make registers, usually *implemented* with D *flip-flops*. A register stores N bits of information, delivering that information in response to a clock pulse. Registers usually have asynchronous set to 1 and clear to 0 capabilities.

Storage Register

A storage register simply stores temporary information, for example incoming information or intermediate results. The size is related to the basic size of information handled by a computer: 8 *flip-flops* for an 8-bit or byte register or 16 bits for a word register. Figure 26 shows a typical circuit and schematic symbols for an 8-bit storage register. Storage registers are important to computer architecture; this topic will be discussed later.

Figure 26: Storage Register



Shift Register

Shift registers also store information and *provide* it in response to a clock signal, but they handle their information differently: When a clock pulse occurs, *instead* of each *flip-flop* passing its result to the output, the *flip-flops* pass their data to each other, up and down the row. For example, in up mode, each *flip-flop* receives the output of the preceding *flip-flop*. A data bit starting in *flip-flop* D0 in a left shifter would move to D1, then D2 and so on until it is shifted out of the register. If a 0 was input to the least significant bit, D0, on each clock pulse then, when the last data bit has been shifted out, the register *contains* all 0s.

Shift registers can be left shifters, right shifters or controlled to shift in either direction.The most general form, a universal shift register, has two control inputs for four *states*: Hold, Shift right, Shift left and Load. Most also have asynchronous inputs for preset, clear and parallel load. The primary use of shift registers is to *convert* parallel information to serial or vice versa. This is useful in interfacing between devices.

Additional uses for a shift register are to *delay* or synchronize data, multiply or divide a number by a factor 2 N or, *provide random* data. Data can be delayed simply by taking advantage of the Hold feature of the register control inputs. Multiplication and division with shift registers is best explained by an example:

Suppose a 4-bit shift register currently has the value $1000_{(2)} = 8_{(10)}$. A right shift results in the new parallel output 0100 = 4 = 8 / 2. A second right shift results in 0010 = 2 = (8 / 2) / 2. Together the 2 right shifts performed a division by 2^2. In general, shifting right N times is *equivalent* to dividing by 2^N. Similarly, shifting left multiplies by 2^N. This can be useful to *compiler* writers to make a computer program run faster.

Random data is *provided* via a ring counter. A ring counter is a shift register with its output fed back to its input. At each clock pulse, the register is shifted up or down and some of the *flip-flops* feedback to other *flip-flops*, generating a *random* binary number. Shift registers with several feedback paths can be used as a pseudo *random* number generator, where the sequence of bits output by the generator meets one or more mathematical criteria for *randomness*.

Applications used on Aircraft systems

We now need to look at how these gates are used in aircraft circuits. You will need to be able to interpret these diagrams and explain how an output is arrived at.

You should be able to describe the operation of a logic circuit if an unexpected logic input is present. Logic gate circuitry is extensively used in aircraft schematic diagrams for all aircraft systems including airframe systems, engine systems, and all avionic systems.

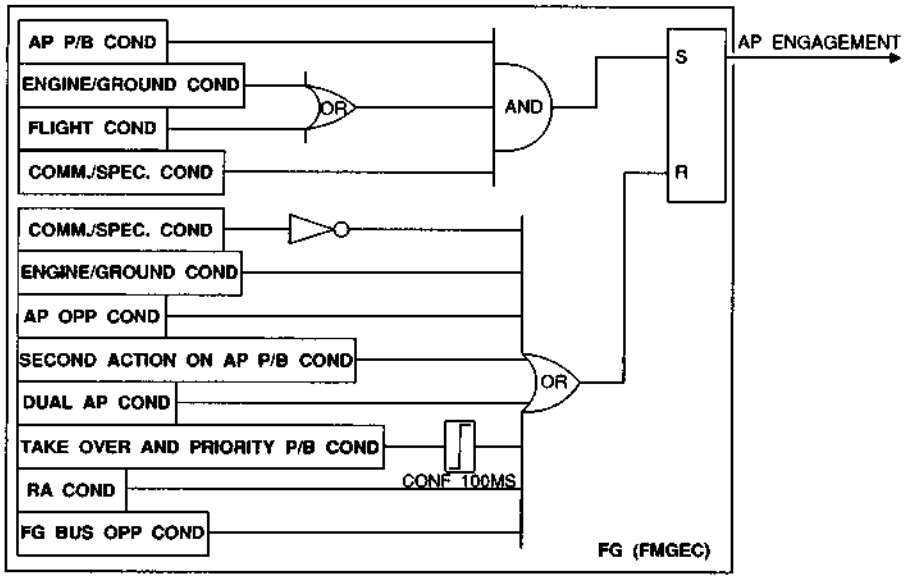
In modern aircraft, most systems are controlled by a *microprocessor* and therefore it is impossible to show all the logic functions, which are in fact performed by a computer. Logic gate circuitry is used today in most cases in the form of simplified diagrams which show the functions that are actually performed within a computer.

The following example in Figure 27 shows the Autopilot Engagement Logic for an Airbus A330. The logic functions shown on this schematic are in fact performed by a so called Flight Management and *Guidance* Computer, but for better understanding of the Autopilot Engagement function, this type of illustration is used.

Activity:

With help of the legend in Figure 27, you should be able to interpret the schematic and explain under which condition the Autopilot can be engaged, or will be disengaged.

Figure 27: Airbus A330 Autopilot Engagement Logic



AP P/ B CONDITION: Autopilot is engaged by action on the engagement pushbutton.
ENGINE/ GROUND CONDITION: Autopilot engagement is possible in any mode if all engines are stopped on ground.
FLIGHT CONDITION: Autopilot engagement is possible 5 seconds after lift- off.
COMMON/ SPECIFIC CONDITIONS: Autopilot engagement is activated if all the components of the system are "healthy".
ENGINE/ GROUND CONDITION: The Autopilot disengages if one engine is started on the ground.
AP OPPOSITE CONDITION: The Autopilot disengages in case of engagement of the opposite Autopilot, if the Autopilot is not in LAND or GO AROUND modes.
SECOND ACTION ON AP P/ B CONDITION: The Autopilot is disengaged by action on the engagement pushbutton, the associated Autopilot being already engaged.
DUAL AP CONDITION: In the event of landing in dual AP operation, in case of a failure the Autopilot will be be disengaged.
TAKE OVER AND PRIORITY P/ B CONDITION:The Autopilot is disengaged by an action on one take over and priority pushbutton switch confirmed during 100 ms.
RA CONDITION: When in landing category CAT 3 DUAL (bi- AP), if Radio Altimeter 1 is lost below 100 feet, AP 1 disengages. The opposite AP is then used because it uses Radio Altimeter 2 (RA 2).
FG BUS OPPOSITE CONDITION:In case of dual AP operation, if the SLAVE FG is not synchronized with the MASTER FG, it forces the disengagement of Autopilot 2.

MD 11 LAMM Schematics

Extensive use of Logic Circuit symbols in Wiring Diagramms is to be found in the LAMM Schematic Manuals of MD11 or MD80 aircraft.

The LAMM Schematic Manuals are named after L. M. LAMM, the originator of the manuals for Mc Donnell Douglas.

The example in Figure 28 shows the Autopilot Engagement Logic on the MD11.

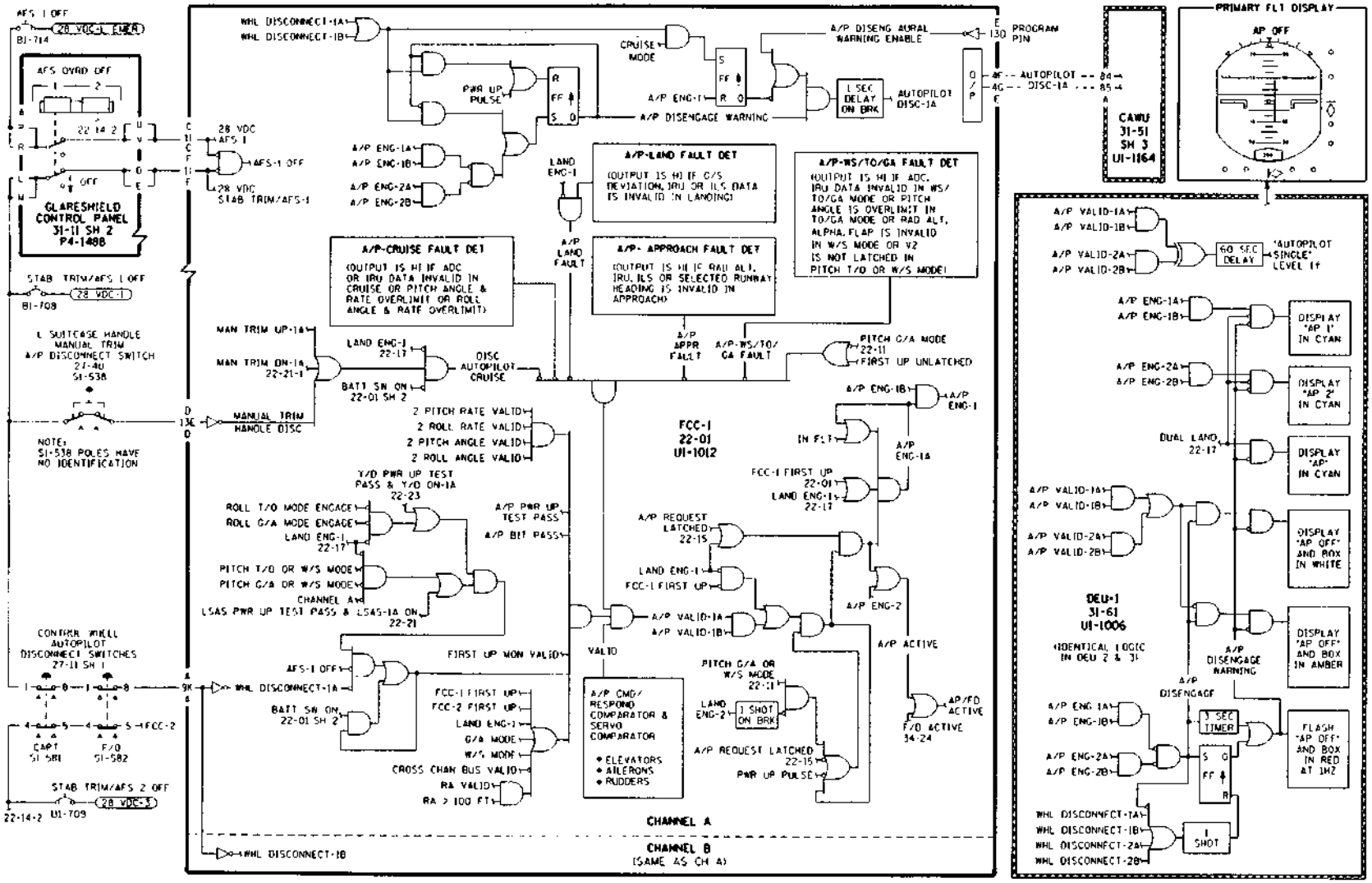
Different manufacturers and different countries have different philosophies on how to draw schematic manuals and wiring diagrams.

As a licensed aircraft mechanic or avionic technician, you should also be able to read and interpret more complicated schematics as those of the MD11.

Activity:

- In the following schematic find the conditions necessary for the engagement of Autopilot 1 and for its disengagement.
- Explain the function of the two Flip Flops in the Autopilot disconnect circuit.

Figure 28: MD11 Autopilot Engagement Schematic



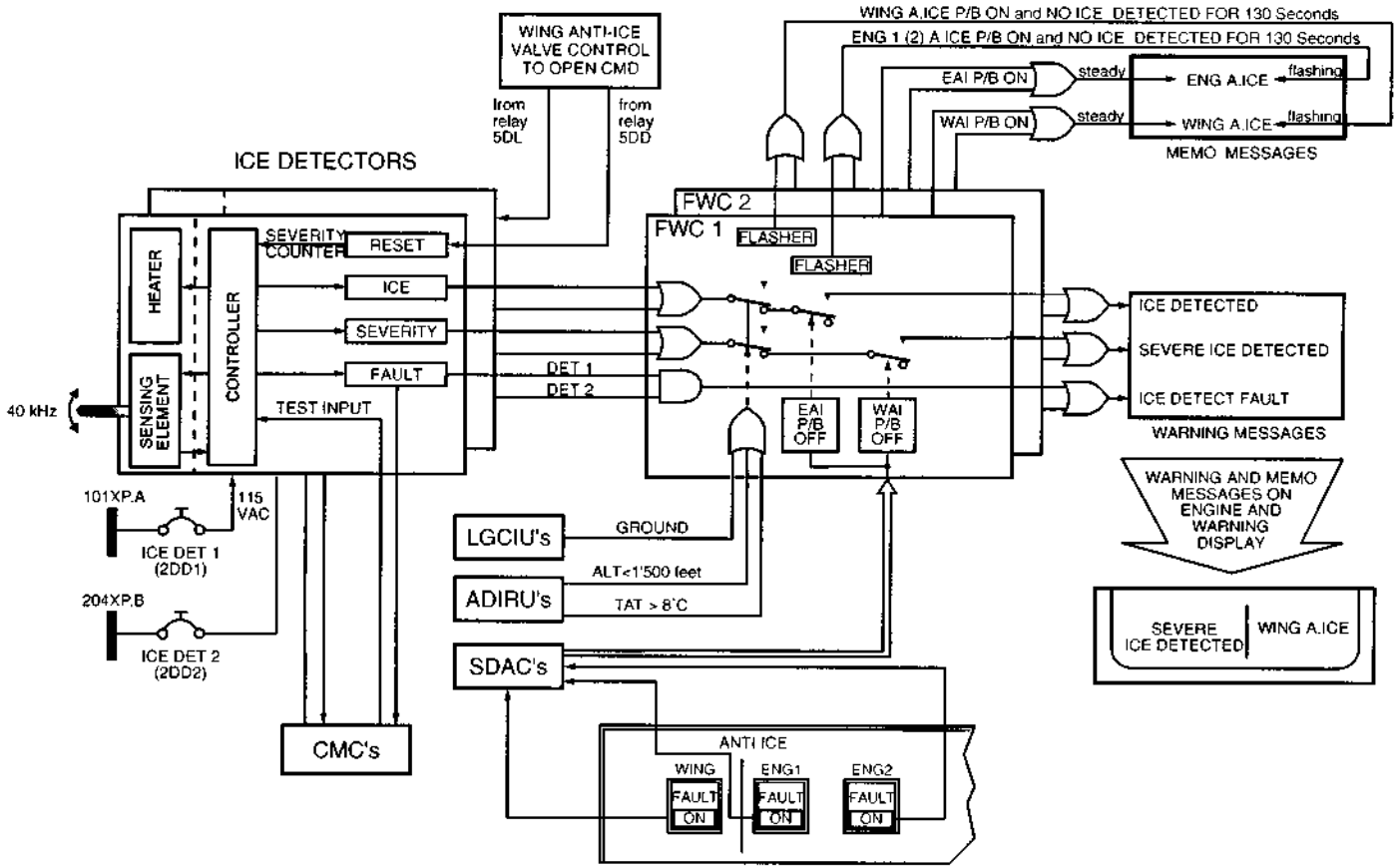
A330 Ice Detection Schematic

This example shows a simplier diagram where the logic for the Ice *Detection* Warnings on the ECAM is shown.

Activity:

Find out what the conditions are, to activate the Ice *detection* warnings, fault messages and memo messages on the ECAM Displays.

Figure 29: A330 Ice Detection Schematic



MD11 APU Starting Circuits Schematic Diagram

On the following pages, two other examples of MD11 LAMM Schematic Diagrams are shown. Figure 30 shows the APU Starting Circuits Schematic and Figure 31shows the Pack 1 Flow Control Valve - Control Schematic.

Activity:

Try to explain the following functions on the two schematic diagrams:

- APU Starting Circuits:
  - Find the signal flow and the necessary logic from pushing the ON/OFF button on the APU Panel, through the Miscellaneous System Controller (MSC), to activate the APU Cont Relay R2-5713 which gives a start signal to the APU ECU.
  - Find the correct logic neccessary to activate the light of the ON/OFF pushbutton on the APU Panel.
  - Find the correct logic neccessary to activate the fuel light on the APU Panel.
  - Find the correct logic neccessary to activate the door light on the APU Panel.
- Pack 1 Flow Control Valve - Control Schematic
  - Explain the conditions which are neccessary to get the Pack 1 0'ride signal in the Environmental System Controller
  - The Pack 1 Override signal is then further used in the Environmental System Controller to close the Pack 1 Flow Control Valve. Explain the neccessary logic conditions.

Figure 30: MD11 APU Starting Circuits Schematic Diagram

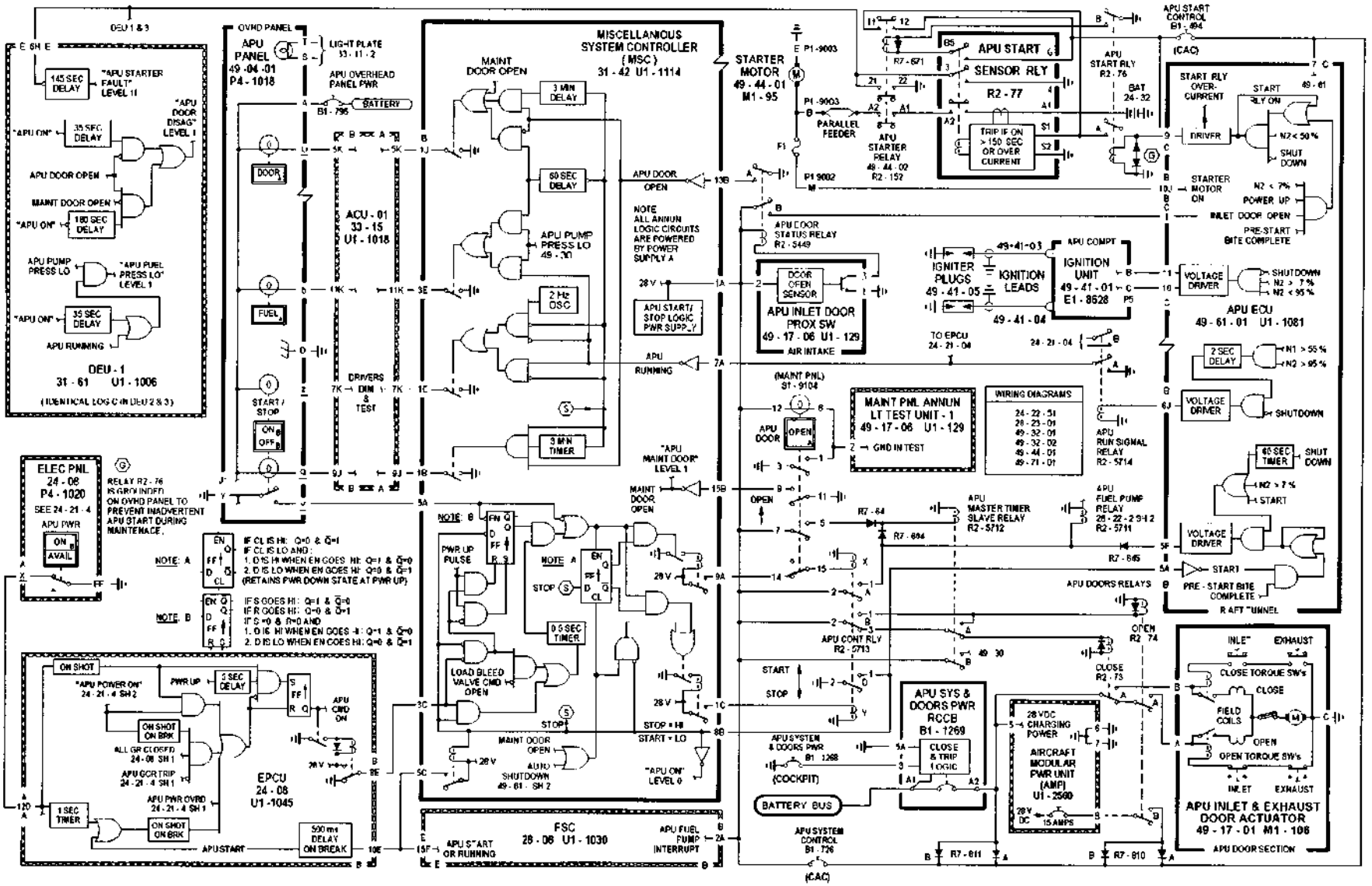
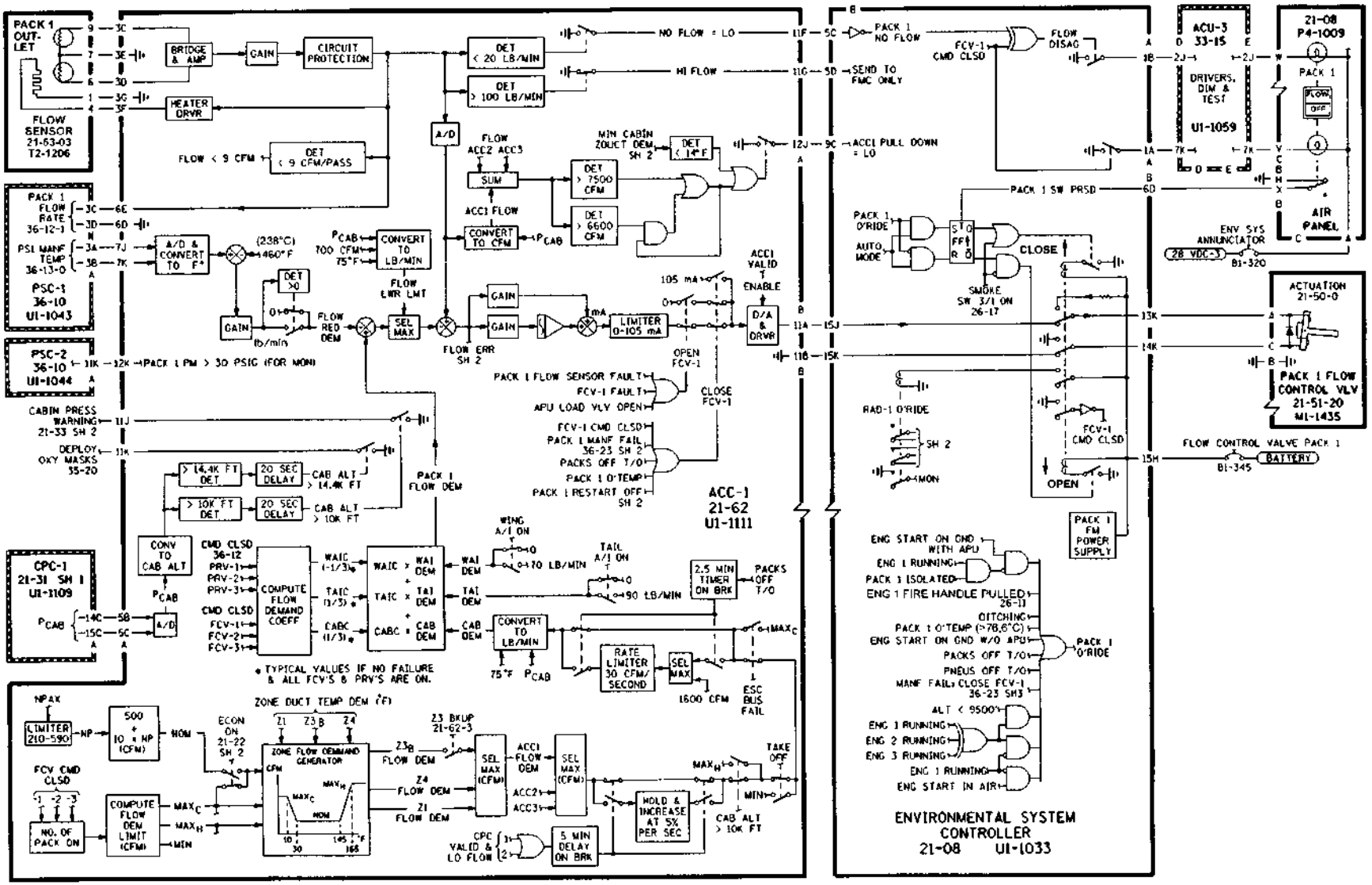


Figure 31: Pack 1 Flow Control Valve - Control Schematic



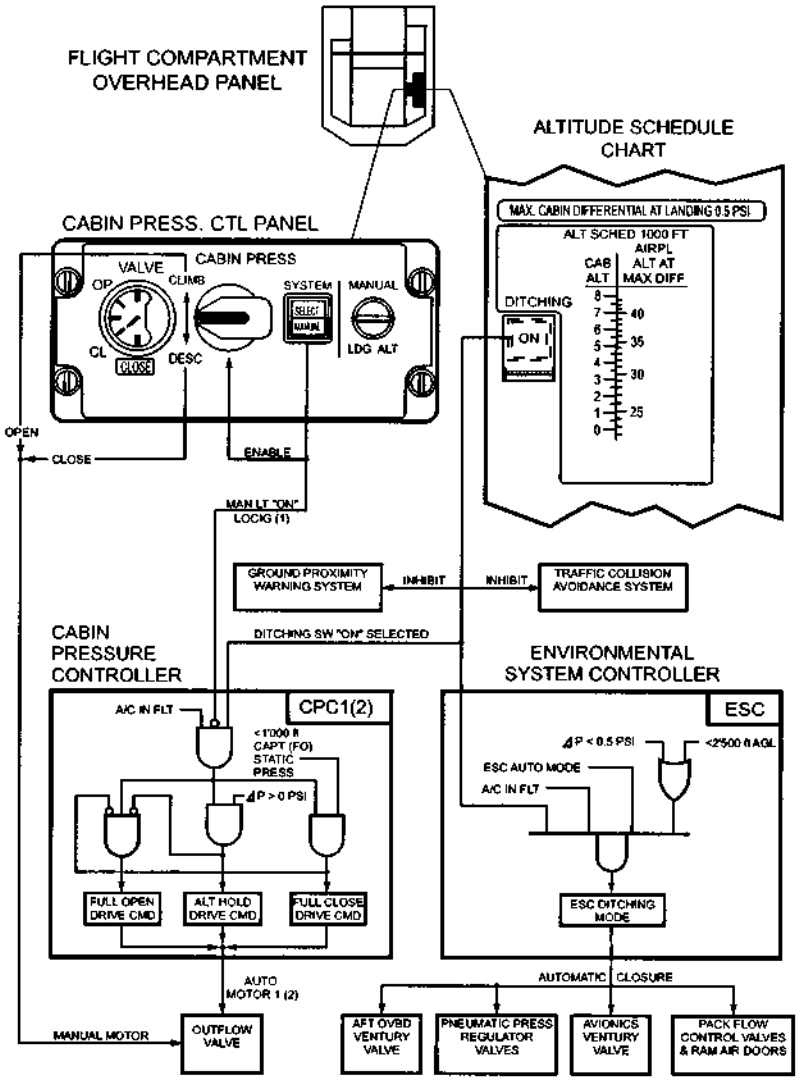
MD11 Ditching Function

This next schematic functional diagram shows the logic which leads to the activation of different functions, when the Ditching Button is pressed on.

Activity:

- Explain the different functions which will follow after the activation of the ditching button.
- Find the correct logic conditions which allow the different functions to become activ.

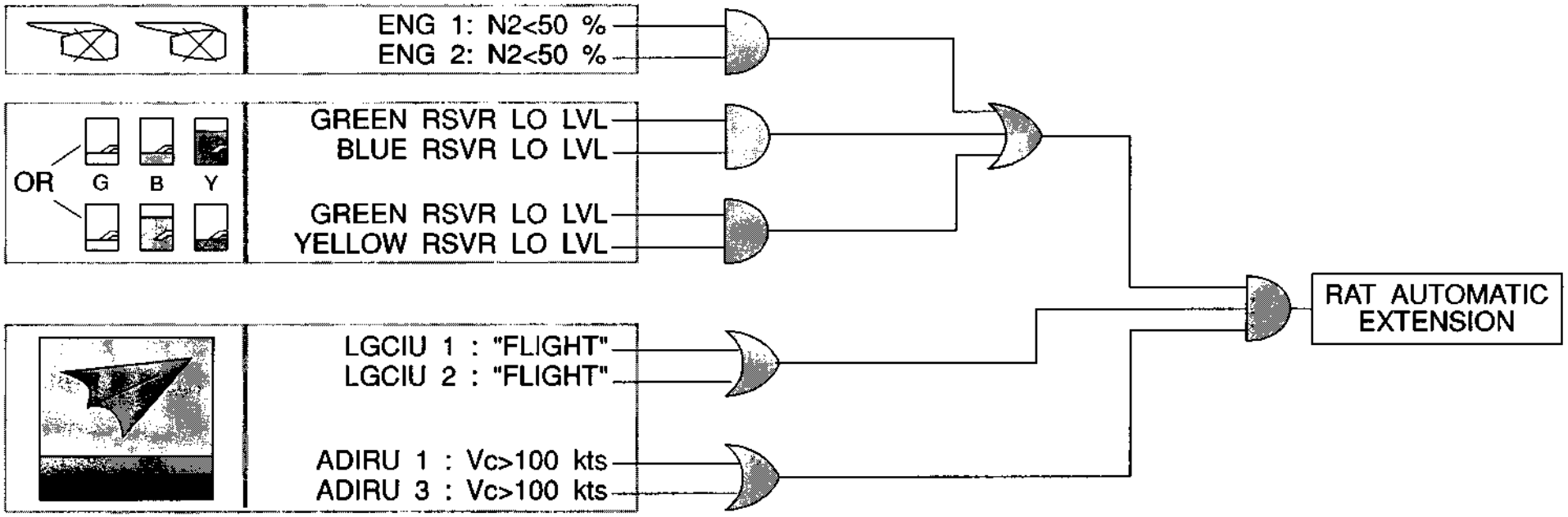
Figure 32: MD11 Ditching Function



Airbus A330 Automatic RAT Extension

Explain the possible causes, which lead to an automatic extension of the Ram Air Turbine:

Figure 33:



5.6 Basic Computer Structure

Introduction to Computers

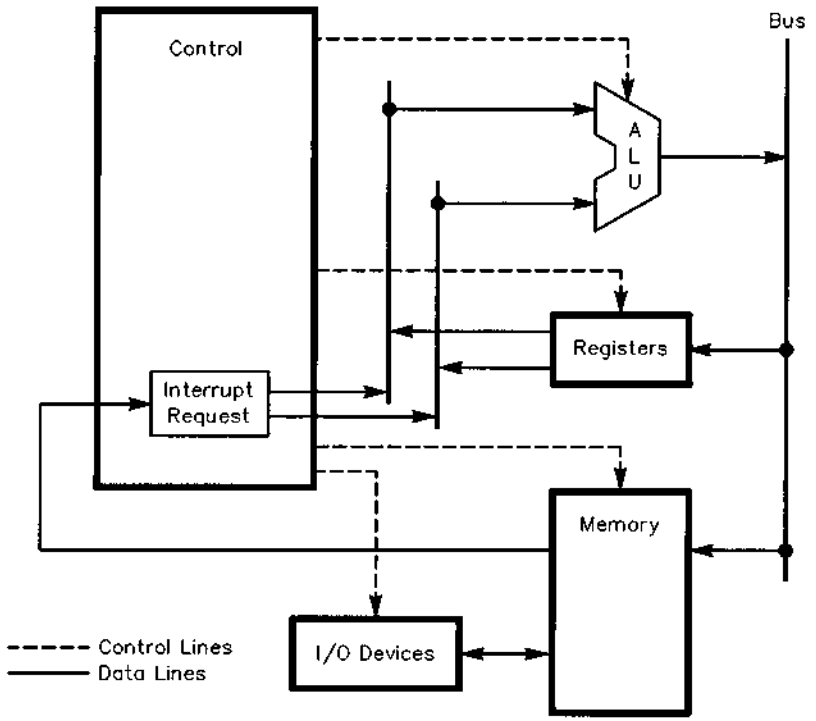
From digital Logic, the *implementation* of that logic with Integrated circuits, interfacing *Integrated Circuit* logic families and the use of memory to store information used by the *IC's*, we come to the computer, which is the synthesis of all these technologies. It combines a *microprocessor IC*, memory chips and user interface into the modern digital computer. A computer combines both components; hardware and a collection of programs; software, to tell it what to do. This sub module will focus on the basic structure of the computer: its internal physical components, the chips and how they work and interact and its external I/O devices for communication with a user.

Computer Organization

The architecture of a computer is the arrangement of its internal subsystems: the *microprocessor*(s), memory, I/O and interfacing. Each subsystem may be concentrated on a single *IC* or spread between many chips. The *microprocessor*, also known as the *central processing unit (CPU)* and usually a single chip, *consists* of three parts: a control unit, an arithmetic logic unit (ALU) and temporary storage registers. A bus — a set of wires to *carry* address, data and control information — interconnects all of the subsystems. Most modern computers are some variation on the basic architecture shown in Figure 1. The *microprocessor*, memory chips and other circuitry are all part of the system's hardware, the physical components of a system. The computer case, the nuts and bolts and physical parts are other parts of the hardware. A computer also includes software, a collection of programs or sequence of instructions to perform a specified task. Some *microprocessors* internally are complete circuitry. The design of general purpose computers is so complex, however, that it is nearly impossible to design an original architecture without any *bugs*. Thus many designers use *microprocessors* that include micro code or micro instructions: instructions in the control unit of a *microprocessor*. This *hybrid* between hardware and software is called firmware. Firmware also includes software stored in ROM or EPROM rather than being stored on magnetic disk or tape.

Computer designers make decisions on hardware, software and firmware *based* on cost *versus* performance. Thus, today's computer market includes a *wide* range of systems, from high-performance super-computers, which cost millions of dollars, to the personal microcomputer, with costs in the thousands new and in the hundreds for older used models.

Figure 1: Basic Computer Structure



The Central Processing Unit

The *central processing unit* is usually a single *microprocessor* chip, although its subsystems can be on more than one chip. The CPU at least includes a control unit, timing circuitry, an arithmetic logic unit (ALU) and also usually *contains* registers for temporary storage.

Control Unit

The control unit directs the operation of the computer, managing the interaction between subunits. It takes instructions from the memory and executes them, performing tasks such as accessing data in memory, calling on the ALU or performing

Input/Output. Control is one of the most difficult parts to design; thus it is the most likely source of *bugs* in designing an original architecture.

Microprocessors consist of both hard wired control and micro programmed control. In both cases, the designer *determines* a sequence of *states* through which the computer cycles, each with inputs to examine and outputs to activate other CPU subsystems (including activating itself, indicating which *state* to do next). For example, the sequence usually starts with "*Fetch* the next instruction from memory," with control outputs to activate memory for a read, a program counter to send the address to be *fetched* and an instruction register to receive the memory contents. Hard wired control is completely via circuitry, usually with a programmed logic *array*. Micro programmed control uses a *microprocessor* with a modifiable control memory, *containing* micro code or micro instructions. An advantage of micro programmed control is flexibility: the code can be changed without changing the hardware, making it easier to correct design errors. Figure 2 and Figure 3 show examples of hard wired control and micro programmed control of a *microprocessor*.

Figure 2: Hard wired Control

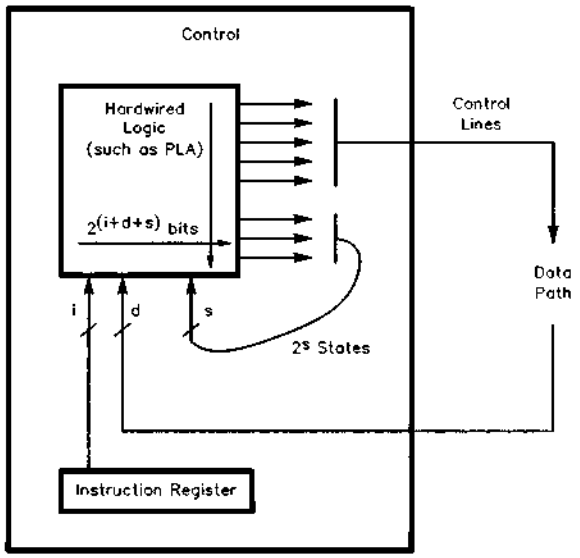
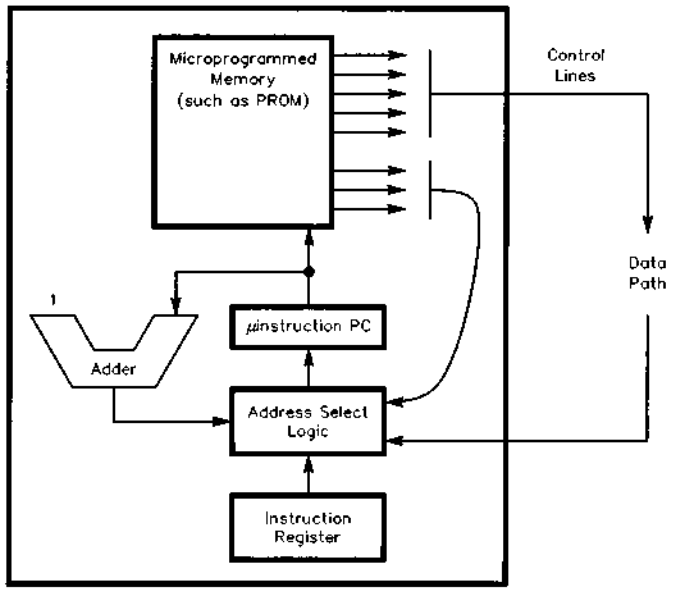


Figure 3: Microprogrammed Control



Timing

Usually, an oscillator controlled by a quartz crystal generates the microcomputer's clock signal. The output of this clock goes to the *microprocessor* and to other *IC*'s. The clock synchronizes the microcomputer subunits. For example, each of the micro instructions is designed to take only one clock cycle to execute, so any components triggered by a micro-instruction's control outputs should finish their actions by the end of the clock cycle. The exception to this is memory, which may take multiple clock cycles to finish, so the control unit repeats in its same *state* until memory says it's done.

Since the clock rate effectively controls the rate at which instructions are executed, the clock frequency is one way to measure the speed of a computer.

Clock frequency, however, cannot be the only criteria *considered* because the actions performed during a clock cycle vary for different designs.

Arithmetic Logic Unit

The arithmetic logic unit (ALU) performs logical operations such as AND, OR and SHIFT and arithmetic operations such as addition, subtraction, multiplication and division. The ALU depends on the control unit to tell it which operation to perform and also to trigger other devices (memory, registers and I/O) to supply its input data and to send out its results to the *appropriate* place.

The ALU often only performs simple operations. Complex operations, such as multiplication, division and operations involving decimal numbers, are performed by dedicated hardware, called *floating-point* processors, or floating point units. These may be included on the original mother board or may be *optional* upgrades. Nowadays in modern microprocessors, usually a floating point unit is integrated on the same microprocessor IC.

Registers

Microprocessor chips have some internal memory locations that are used by the control unit and ALU. Because they are inside the *microprocessor IC*, these registers can be accessed more quickly than other memory locations. Special purpose registers or dedicated registers are purely internal, have *predefined* uses and cannot be directly accessed by programs. General purpose registers hold data and addresses in use by programs and can be directly accessed, although usually only by assembly *level* programs.

The dedicated registers include the instruction register, program counter, effective address register and status register. The first step to execute an instruction is to *fetch* it from memory and put it in the instruction register (IR). The program counter (PC) is then *incremented* to *contain* the address of the next instruction to be *fetched*. An instruction may change the program counter as a result of a conditional *branch* (if-then), *loop*, subroutine call or other non linear execution. If data from memory is needed by an instruction, the address of the data is calculated and *fetched* with the effective address register (EAR).

The status register (SR) keeps *track* of various conditions in the computer. For example, it tells the control unit when the keyboard has been typed on so the control unit knows to get input. It also notices if something goes wrong during an instruction execution, for example an attempted divide by 0, and tells the control unit to halt the program or fix the error. Certain bits in the status register are known as the condition codes, flags set by each instruction. These flags tell information about the result of the latest instruction — such as if the result was negative or positive or zero and if an arithmetic overflow or a *carry* error occurred. The flags can then be used by a conditional *branch* to decide if that *branch* should be taken or not.

Some architectures also use a *stack* pointer (SP) and/or an accumulator. In a *stack* system, a memory location is designated as the "bottom" of the *stack*. Data to be stored is always added to the next memory location, the "top" of the *stack*; and data to be accessed must always be taken off the "top." (This technique is called "last in, first out," or LIFO.)

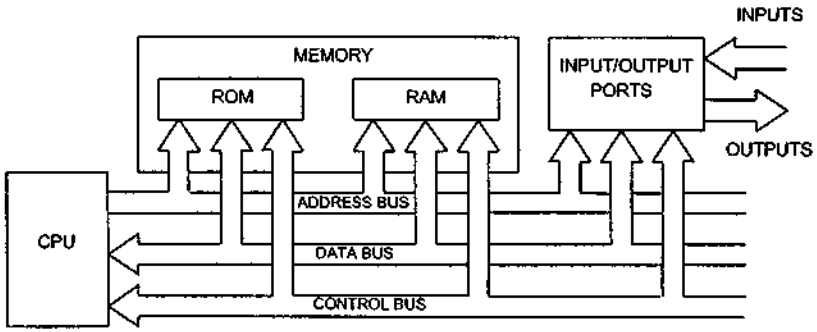
Highway Structure

Figure 4 shows the highway structure, there are three buses, the address bus, the data bus and the control bus. As we have already seen the memory *consists* of a number of locations, each individually identified by an address.

The address bus is therefore used to specify the memory location or input/output port involved in the transfer. It is a one way bus and may have anything from 4 to 64 lines depending on the number of memory addresses there are, 8 lines give 2^8=256 addresses. The data bus is a bi-directional bus and is used to *carry* the data being transferred to and from the memory or input and output transfer.

The control bus *comprises* input and output lines which synchronise the *microprocessor's* operation with that of the external circuitry i.e. read/write controls, timing signals, input/output selection. This is also a bi-directional bus.

Figure 4: Highway Structure



Memory

Computers and other digital circuits rely on stored information, either data to be acted upon or instructions to direct circuit actions. This information is stored in memory devices, in binary form. This section first discusses how to access an individual item in memory and then compares different memory types, which can vary how quickly and easily an item is accessed.

Accessing a Memory Item

Memory devices *consist* of a large number of memory cells each capable of remembering one bit of binary information. The information in memory is stored in digital form with collections of bits, called words, representing numbers and symbols. The most common symbol set is the American National Standard Code for Information Interchange (ASCII). Words in memory, just like the letters in this sentence, are stored one after the other. They are accessed by their location or address. The number of bits in each word, equal to the number of memory cells per memory location, is constant within a memory device but can vary for different devices. Common memory devices have word sizes of 8, 16 and 32 bits.

Addresses and Chip Size

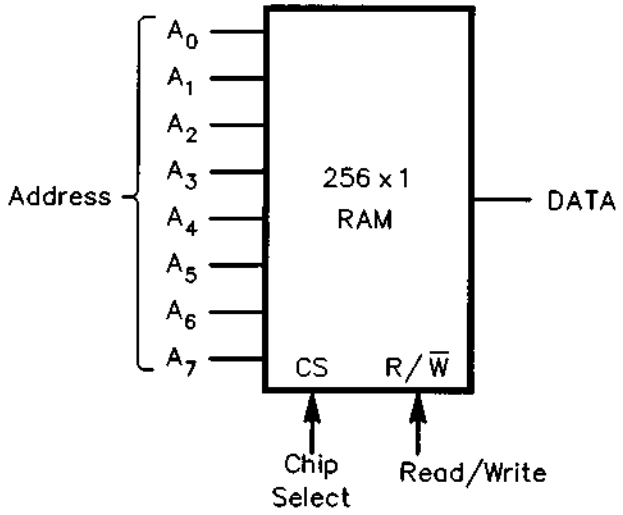
An address is the identifier, or name, given to a particular location in memory. Since this address is expressed as a binary number, the number of *unique* addresses available in a particular memory chip is *determined* by the number of bits to express the address. For example, a memory chip with 8 bit addresses has 2^8 = 256 memory locations. These locations are accessed as the addresses 00000000 through 11111111, 0 through 255 decimal or 00 through FF hex. (For ease of notation, programmers and circuit designers use hexadecimal, *base* 16, notation to avoid long *strings* of 1s and 0s.) The memory chip size can be expressed as M × N, where M is the number of *unique* addresses, or memory locations and N is the word size, or number of bits per memory location. Memory chips come in a variety of sizes and can be arranged, together with control circuitry and decoders, to meet a designer's needs.

Basic Structure

Memory chips, no matter how large or small, have several things in common. Each chip has address, data and control lines, as shown in Figure 5. A memory chip must have enough address lines to uniquely address each of its words and as many data lines as there are bits per word. For example, the 256 × 1 memory has 8 address lines and 1 data line.

The control lines for a memory chip can vary. The figure shows a simple example: two control lines, a W / R and CS. In this case, data lines transfer both inputs (when writing) and outputs (when reading) so the W / R control line is needed to put the memory chip in read mode or write mode. The chip select, CS, control line tells the chip whether it is in use. When the chip is selected, it is "on," acting upon the address, data and W / R information presented to it. When the chip is not selected, the data line enters a high-*impedance state* so that it does not affect, and is not affected by devices or circuits attached to it.

Figure 5: Example of a 256 × 1 memory chip.



Reading and Writing

To write (store data in) or read (retrieve data from) a memory device, it is necessary to *gain* access to specific memory cells. A small 256 × 1 memory chip is used as an example. Later, this example will be expanded to a larger computer memory system.

If we want to write a 1 to the 11th word of the 256 × 1 memory (such as memory location 10 decimal or 00001010 binary), we must execute the following steps:

1. Place the correct address (00001010) on the address lines.
2. Place the data to be written (1) on the data line.
3. Set the W / R control line to write (low, 0).
4. Set the CS control line to select (high, 1). (Many memory devices use an active low chip select, $\overline{CS}$.).

This writes the data on the data line (1) to the address on the address lines (00001010).

The steps to read the contents of the 11th word are similar except that the W / R control line is set to read (high, 1).

Timing

Accurate timing *requirements* must also be incorporated into the above steps. While writing, the address and data information must be present for a minimum setup time before and hold time after, the CS and signals have been activated. This is to avoid *spurious* signals *spraying* all over the memory *array*. While reading, address line changes are not harmful, but the output data is only valid a minimum access time after the last address input is stable. Manufacturers' data sheets and application notes *provide* the timing specifications for the particular *IC* you are using.

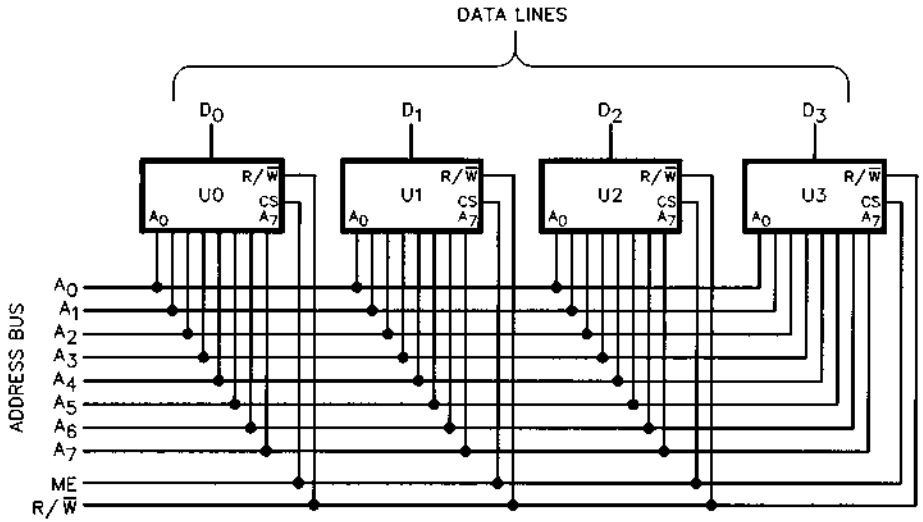
Larger Words

The one-bit-*wide* memory described above *provides* a good introduction, but usually we want a *wider* memory. One way to get *wider* memory is to use several 1-bit-*wide* memory chips. The address and control lines go to each chip, and data from each chip is used as a single bit in the large word. It is easy to see that when reading from address 0A (hex), the data lines D0 through D3 *contain* the data from address 0A of chips U0 through U3.

An address placed on the shared address lines (called an address bus) now specifies an entire word of data. Notice that one line of the address bus connects to the CS pin of each memory chip. This line is labelled ME, or memory enable, and sets all four *IC*'s active at the same time, in order to read or write data when the R/W is activated additionally.

If all four memory chips were put in a single package, they would make a 256 × 4 *IC*. This *IC* would look like the chip in Figure 5, except that it would have 4 data lines.

Figure 6: A 256 × 4 memory built with four 256 × 1 memory chips.



More Address Space

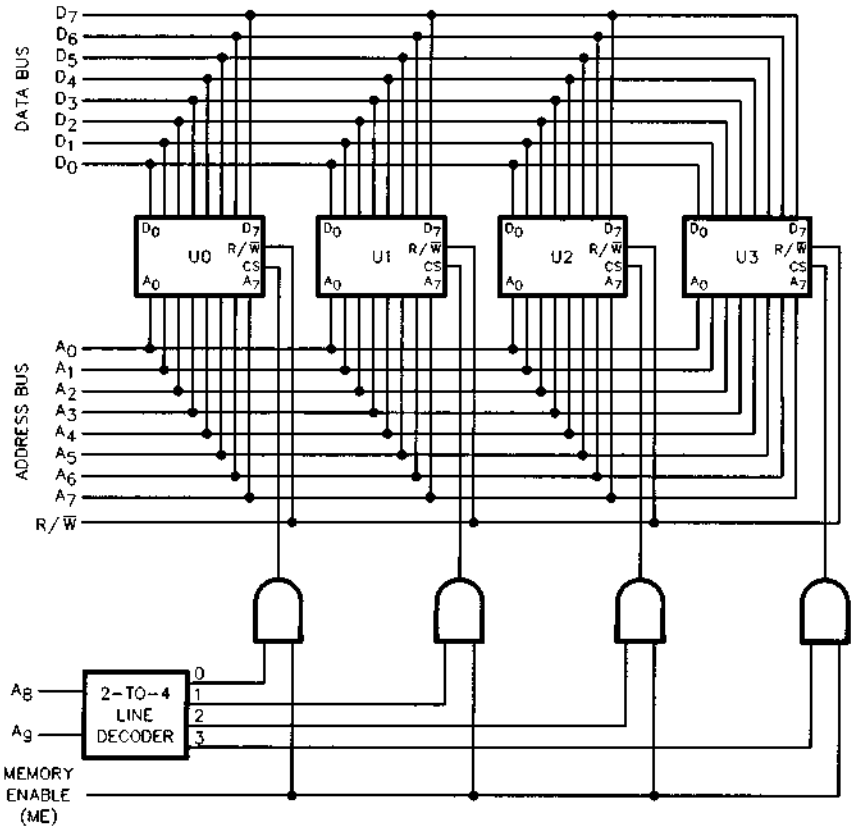
For even larger memory systems, the same principles as shown in Figure 7 can be applied. Figure 7 shows a 1024 × 8 (1 kilobyte) memory built from four 256 × 8 memory chips. A kilobyte is usually abbreviated as K. Notice this is not quite the same as the metric prefix kilo, because it represents 1024, rather than 1000.

Ten address lines are needed to address 1024 locations (2^{10} = 1024). Eight of the 10 address lines, A0 to A7, are used as a normal address bus for chips 0 through 3. The remaining 2 address lines, A8 and A9, are run through a 2-to-4 line decoder to choose between the 4 memory chips. When *employed* in this manner, the 2-to-4 line decoder is called an address decoder.

To assert the CS input for one of the memory chips, ME must be 1 and the correct output of the 2-to-4 line decoder must also be 1. When an address is placed on A0 through A9, a single memory chip is selected by ME, A8 and A9. The other 8 address lines address a single word from that chip. The three chips that are not selected enter a high-*impedance state* and do not affect the data lines. This example

shows that, using the proper memory chips and address decoding, any size memory with any word length can be built.

Figure 7: A 1024 × 8 memory built with four 256 × 8 memory chips

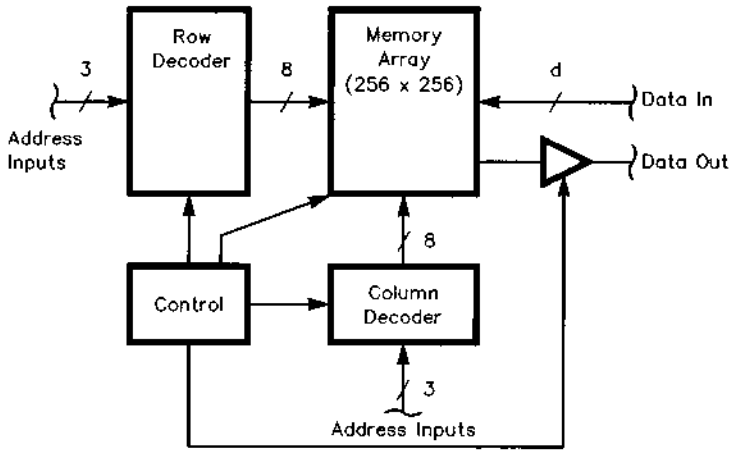


Alternate Memory Structures

Figure 8 shows how the same chip can be accessed in different ways by using two decoders, a row decoder and a column decoder. The same 256 × 256 memory *array* can be treated as a 64 K × 1 *array*, a 256 × 256 *array* or other possibilities. In fact, most larger memory chips are made as square *arrays*: 32 × 32 (1024 bytes or 1 K), 64 × 64 (4096 bytes or 4 K), 256 × 256 (65536 bytes or 64 K), 1024 × 1024 (1 M), 2048 × 2048 (4 M), 4096 × 4096 (16 M) and so on. (Here the M represents a mega-byte, which is 1048576 bytes.)

The square *array* makes the chips more cost effective to manufacture (easier quality control and less waste) and easier to incorporate into a printed-circuit-board circuit layout. Notice that each M × N is a *power of* 2. So while we refer to the chips by shorter names like 1 Mbyte, the actual number of memory cells is larger than 1000000. The product, M × N, only refers to the number of memory cells in the chip; and designers are free to choose the word size *appropriate* to their needs. In fact, they may access one location as an 8-bit word and another as a 16-bit word. For example, a computer with a Motorola MC68000 *microprocessor* automatically accesses a character, such as "A," as 8 bits (a byte), an integer as 16 bits (a word) and a real value as 32 bits (a longword). (Apple Macintosh computers use the MC68000 *microprocessor*.) To further complicate things, a different manufacturer may call 16 bits a halfword and 32 bits a word. In using memory, the controller chips and circuitry to access the memory can be just as important as the memory itself.

Figure 8: Row and column decoders



Memory Types

The concepts described above are applied to several types of *random*-access, *semiconductor* memory. *Semiconductor* memories are categorized by the ease and speed with which they can be accessed and their ability to "remember" in the absence of power.

Volatile Memory

SAM versus RAM

Sequential-access memory (SAM) must be accessed by stepping past each memory location until the desired location is reached. Magnetic tapes *implement* SAM; to reach information in the middle of the tape, the tape head must pass over all of the information on the beginning of the tape. Two special types of SAM are the *queue* and the push-down *stack*. In a *queue*, also called a first-in, first-out (FIFO) memory, locations must be read in the order that they were written. The *queue* is a "first-come, first-served" device, like a line at a ticket window. The push-down *stack* is also called last-in, first-out (LIFO) memory. In LIFO memory, the location written most recently is the next location read. LIFO can be visualized as a *stack*, always adding to and removing from the "top" of the *stack*.

Random-access memory (RAM) allows any memory cell to be accessed at any instant, with no time wasted stepping past the "beginning" parts of the data. *Random*-access memory is like a bookcase; any book can be pulled out at any time.

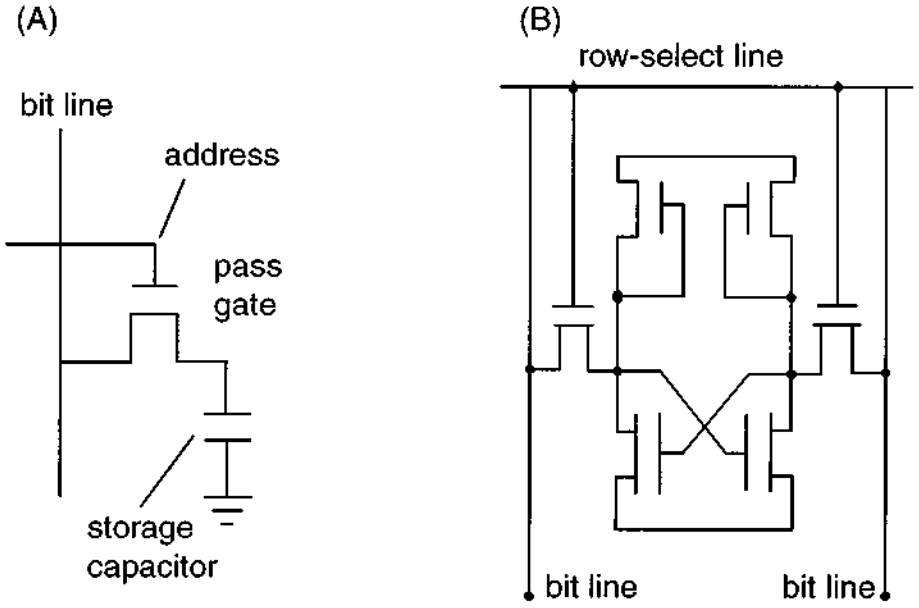
It is usually faster to access a desired word in RAM than in SAM. Also, all words in RAM have the same access time, while each word in a SAM has a different access time *based* on its position. Generally, the *semiconductor* memory devices internal to computers are *random*-access memories. Magnetic devices, such as tapes and disks, have at least some sequential access characteristics. We will leave tapes and disks for a later section and concentrate here on *random*-access, solid-*state* memories.

Random Access Memory

Most RAM chips are *volatile*, meaning that stored information is lost if power is removed. RAM is either static or dynamic. ***Dynamic RAM (DRAM)*** stores a bit of information as the presence or absence of charge. This charge, since it is stored in a capacitor, slowly leaks away. It must be refreshed periodically. Memory refresh typically occurs every few milliseconds and is usually performed by a dynamic RAM controller chip. ***Static RAM (SRAM)*** stores a bit of information in a *flip-flop*. Since the bit will retain its value until either power is removed or another bit replaces it, refresh is not necessary.

Both types of RAM have their advantages and disadvantages. The advantage of DRAM is increased *density* and ease of manufacture, making them significantly less expensive. SRAM's, however, have much faster access times. Most general purpose computers use DRAMs, since large memory size and low cost are the major objectives. Where the amount of memory required doesn't *justify* the use of DRAM, and the faster access time is important, SRAM's are common, for example, in *embedded* systems (telephones, toasters) and for *cache* memories. Both types of RAM are available in MOS families; SRAM's are also available in bipolar. Generally, MOS RAMs have lower power consumption than bipolar RAMs, while access speeds vary *widely*. Cost, power consumption and access time, *provided* in manufacturers' data sheets, are factors to *consider* in selecting the best RAM for a given application.

Figure 9: (A) DRAM Cell vs. (B) an SRAM Cell



Non-Volatile Memory

Read-Only Memory (ROM)

Read-Only Memory is *nonvolatile*; its contents are not lost when power is removed from the memory. *Despite* its name, all ROM's can be written or programmed at least once. The earliest ROM designs were "written" by clipping a diode between the memory bit and power supply wherever a 0 was desired. Modern MOS ROM's use a transistor *instead* of a diode. Mask ROM's are programmed by having ones and zeros *etched* into their *semiconductors* at manufacturing time, according to a pattern of connections and non-connections *provided* in a mask. Since the "programming" of a mask ROM must be done by the manufacturer, adding expense and time delays, this type of ROM is primarily used only in high volume applications.

PROM

For low volume applications, the programmable ROM (PROM) is the most effective choice since the data can be written after manufacture. A PROM is manufactured with all its diodes or transistors connected. A PROM programmer device then "burns away" undesired connections (see Figure 10). This type of PROM can be written only once.

EPROM, EEPROM

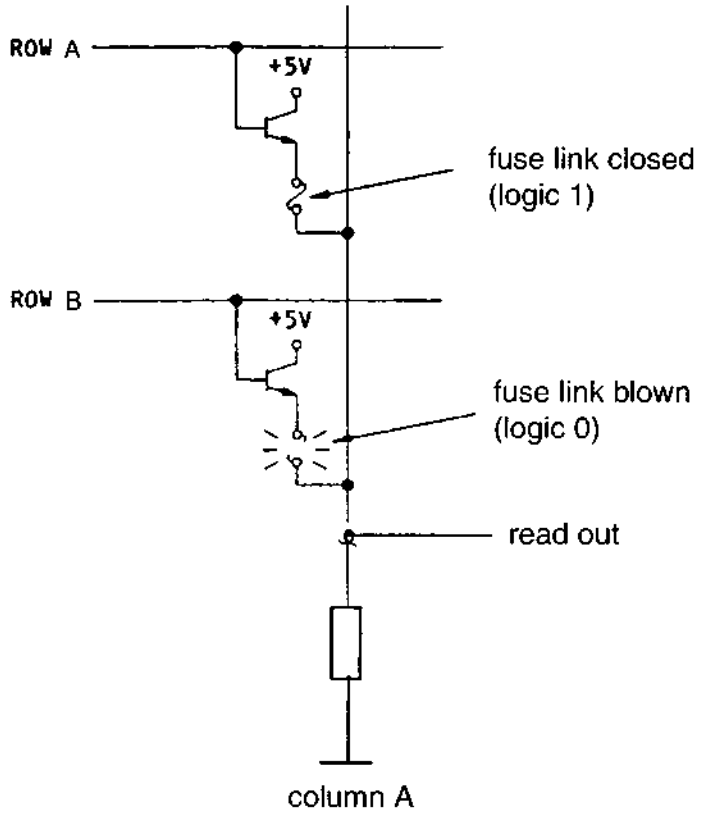
Two types of PROMs that can be "erased" and reprogrammed are EPROM's and EEPROM's. The transistors in UV *erasable* PROMs (EPROM's) have a *floating* gate surrounded by an insulating material. When programming with a bit value, a high voltage creates a negative charge on the *floating* gate. Exposure to ultraviolet light erases the negative charge. Similarly, electrically *erasable* PROMs (EEPROM's) erase their *floating*-gate values by *applying* a voltage of the opposite polarity.

ROM's are practical only for storing data or programs that do not change frequently and must survive when power is removed from the memory. The programs that start up a computer when it is first switched on or the memory that holds the call sign in a repeater are prime candidates for ROM.

Flash

Flash also known as FEPROM is one of the newest types of *nonvolatile* memory. In this device data is erased and reprogrammed in blocks, unlike the byte by byte altering capability of EEPROM. FLASH memory has the *density* of EPROM's and the electrical erase capability of standard EEPROM's. FLASH is used in digital cameras to store pictures, PC's store their BIOS on FLASH memory chips, and MP3 audio recorders use FLASH to store music.

Figure 10: Programmable ROM



Nonvolatile RAM

For some situations, the ideal memory would be as *nonvolatile* as ROM but as easy to write to as RAM. The primary example is data that must not be allowed to *perish despite* a power failure. Low-power RAMs can be used in such applications

if they are supplied with NiCd or lithium cells for backup power. A more elegant and durable solution is *nonvolatile* RAM (NVRAM), which includes both RAM and ROM. The standard *volatile* RAM, called shadow RAM, is backed up by *nonvolatile* EEPROM. When the RECALL control is *asserted*, such as when power is first applied, the contents of the ROM are copied into the RAM. During normal operation, the system reads and writes to the RAM. When the STORE control is triggered, such as by a power failure or before turning off the system, the entire contents of the RAM are copied into the ROM for *nonvolatile* storage. In the event of primary power failure, to successfully save the RAM data, some power must be maintained until the memory store is complete (+5 V for 20 ms).

Cache versus Main Memory

Memory is in high demand for many applications. To balance the *trade-off* of speed *versus* cost, most computers use a larger, slower, but cheaper main memory in *conjunction* with a smaller, faster, but more expensive *cache* memory. As you run a computer program, it accesses memory frequently. When it needs an item, a piece of data or the next part of the program to execute, it first looks in the *cache*. If the item is not found in the *cache*, it is copied to the *cache* from the main memory. As you run a computer program, it often repeats certain parts of the program and repeatedly uses pieces of data. Since this information has been copied to the high-speed *cache*, your computer game or other application can run faster. Information used less often or not being used at all (programs not currently being run) can stay in the slower main memory.

A "*cache*" is a place to store *treasure*; the *treasure*, the information you are using frequently, can be accessed quickly because it is in the high-speed *cache*. The use of *cache versus* main memory is managed by a computer's CPU so it is transparent to the user. The *improvement* in program execution time is similar to accessing a floppy disk *versus* the computer's internal memory.

Peripherals

Peripherals are any devices outside the CPU. They *provide* additional capabilities. One of the most common examples being communication with a user via input devices and output devices. Input devices *provide* the computer both data to work on and programs to tell it what to do. Output devices present the results of computer operations to the user or another system and may even control an external system.

Both input and output combine to *provide* user friendly interaction. This section discusses the most common user interfaces.

Most of these devices have adapted to certain standards and use readily available connection cables. Thus, they can be easily incorporated into a system, and a knowledge of the internal actions is not necessary. A knowledge of how external memory devices work is more useful and will be discussed in more detail.

Input Devices

The keyboard is probably the most familiar input device. A keyboard simply makes and breaks electrical contacts. The open or closed contacts are usually sensed by a *microprocessor* built into the circuit board under the keys. This *microprocessor* decodes the key closures and sends the *appropriate* ASCII code to the main computer unit. Keyboards will generate the entire 128 character ASCII set and often, with CONTROL and ALT (Alternate) keys, the 256 character extended ASCII set.

The mouse is becoming increasingly popular for use with graphical user interfaces. The mouse casing holds a ball and circuitry to act as a multidirectional detection device. By moving the mouse, the ball rolls, controlling the relative position of a cursor on the screen. Buttons on the mouse make and break connections (clicking) to select and activate items (icons) on the screen. The trackball is a variation of the mouse.

Other input devices include modems and magnetic disks and tapes. Magnetic disks and tapes, discussed at length later, *provide* additional external memory. Newer input devices include voice activated devices, touch screens and scanners.

Output Devices

The most familiar output device is the computer screen, or monitor. For smaller character displays, LED *arrays* can be used. The next most common output device is the printer, to produce paper hard copy. Modems and magnetic disks and tapes are output devices as well as input devices. Newer output devices include speech synthesizers. The output devices (except the sound device) share a common display technique: images, such as characters and graphics, are formed by tiny *dots*, called *pixels* (picture elements). On screens, these are *dots* of light turned on and off. In printers, they are *dots* of ink imposed onto the paper. For colour displays, *pixels* in red, green and blue (RGB) are spaced closely together and appear as numerous colours to the human eye.

Video monitors are usually specialized cathode-ray tube (CRT) displays. In newer notebook computers, the monitors are being fabricated with monochrome or colour *liquid-crystal displays* (*LCD*'s), but colour *LCD* displays are still quite expensive. A standard TV set can even be used as a computer monitor. Two techniques are used to turn on the screen *pixels*. Raster scanning covers the screen by writing one row of *pixels* at a time, from left to right and top to bottom. Then, a vertical *retrace* brings the beam back to the top of the screen to begin again. Raster scanning signals every *pixel* on or off for each screen pass. An alternative, vector mode, only signals the *pixels* where something on the screen has changed.

Light-*emitting* diodes (LED's) are handy when a full screen display is not necessary. A single diode easily indicates on/off *states* such as the power light on many devices. A popular single character display is the seven segment version, which is good for displaying numbers.

Disk Drives

Magnetic Disc Drives

Magnetic media are essential input/output devices since they *provide* additional memory. The earliest ways to store programs and data were on *punched* cards and tape. Some early home computers used audio cassettes.

Disk storage is *prevalent* when *random* access is needed. In some ways, disk storage is similar to that of a record player. The data is stored in circles (tracks) on a round platter (disk or diskette) and accessed by a device (a head) moving over the platter. Unlike the record player, the *tracks* are concentric rather than spiral and the head can write as well as read.

Figure 11A shows an example of the disk recording surface. Usually the *tracks* are divided into equal-sized storage units, called sectors. Also, since the disk has two sides, most disks can store information on both sides. Therefore, locating a piece of information in disk memory means identifying three coordinates: the side, *track* and sector.

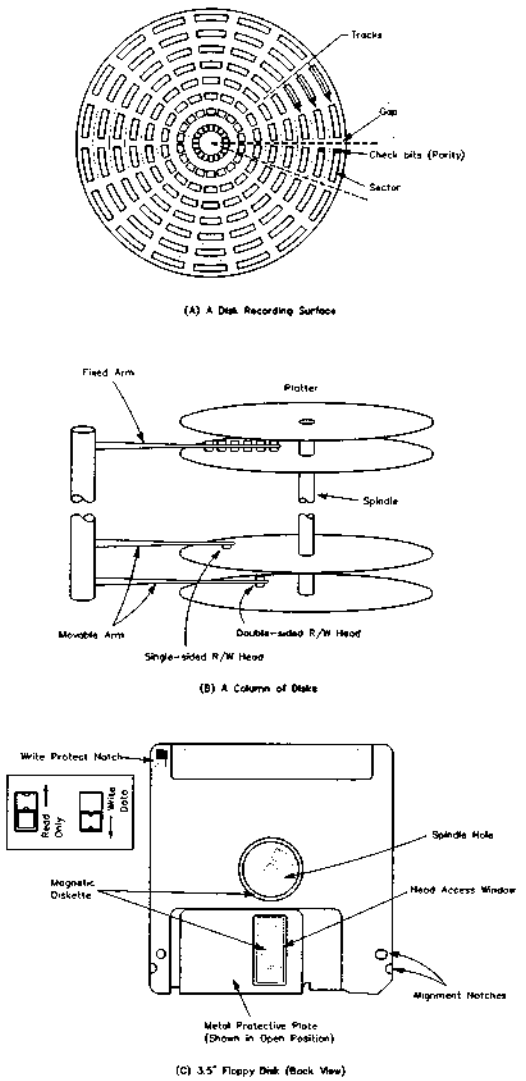
Accessing a piece of information on the disk system involves a number of wait times until the data access is complete. First, disks may be either movable-head or fixed-head, as shown in Figure 11B. In the movable system, a single read/write

head is attached to a movable arm, so there is a seek time for the movable arm to position the read/write head on the *appropriate track*. In a fixed system, each *track* has its own read/write head, so seek time is zero since the head is immediately in position. Second, the data must rotate into position under the read/write head. This time is called *latency*. Finally, there is the normal time for the read/write to *occur*. A number of types of disk technology are available. In hard disk systems, the disk is *rigid* and the read/write head does not contact the disk directly. The absence of *friction* between the head and disk allows finer head positioning and higher disk speeds. Thus, hard disks hold more data and are accessed more quickly than floppy disks. Floppy disks enclose the magnetic-media platter in a casing, as shown in Figure 11C, so the disk can be carried around. The floppy disk can be inserted into a disk drive and the read/write head automatically extended; when done, the read/write head is automatically retracted before the disk is ejected from the drive. Variations in floppy disks include single-sided (SS) or double-sided (DS); single, double or high *density*; and 3 1 /2. The *density* refers to the disk format used by the disk controller. High data *density* allows more data to be written to the disk but requires a higher quality diskette. Not all disks can be written as high *density* and not all disk drives can read high *density* disks.

Dust and dirt on the disk and the imperfections in the disk surface gradually damage both the disk and the head. This means that disks *eventually* wear out, and the data on the disk will probably be lost.

Therefore, it is prudent to make backup copies of your disks, stored in a clean, dry, cool place.

Figure 11: Magnetic Disk Drives



Optical Disks

Another form of memory is the optical disk, which uses optical rather than electrical means for reading and writing. It developed from videodisc technology during the early 1980s. Optical disks have a greater memory capacity than most magnetic disks; the largest ones can store 1.5 gigabytes of information, which is equal to about 700,000 pages of printed material. Optical disks come in sizes ranging from 3.5 to 12 inches (30 cm). They are *widely* used as auxiliary memory when large memory capacity is required.

CD-ROM

CD-ROM is an abbreviation of Compact Disc Read-Only Memory, a type of computer memory in the form of a compact disc that is read by optical means. A CD-ROM drive uses a low-power *laser* beam to read digitized (binary) data that has been encoded in the form of tiny *pits* on an optical disk. The drive then feeds the data to a computer for *processing*.

The standard compact disc was introduced in 1982 for digital audio reproduction. But, because any type of information can be represented digitally, the standard CD was adapted by the computer industry, beginning in the mid-1980s, as a low-cost storage-and-distribution medium for large computer programs, graphics, and databases. With a storage capacity of 680 megabytes, the CD-ROM found rapid commercial acceptance as an alternative to so-called floppy disks (with a maximum capacity of 1.4 megabytes).

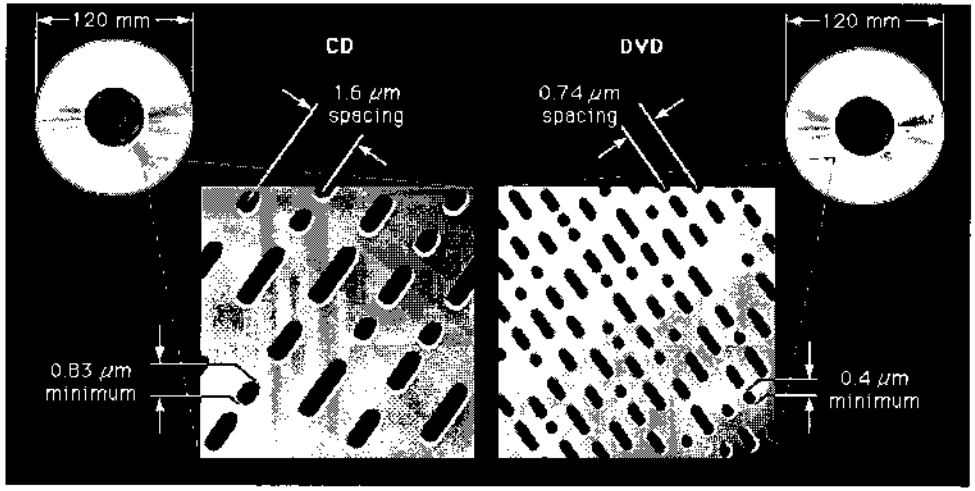
Unlike conventional magnetic storage technologies (e.g., tapes, floppy disks, and hard disks), CD's and CD-ROM's are not recordable--*hence* the tag "read only." This limitation *spurred* the *development* of various recordable magnetic-optical *hybrid* storage devices; but they generally failed to penetrate *beyond* the publishing world, where large multimedia files are regularly exchanged, because of in*compatibility* with standard CD and CD-ROM players. In the early 1990s a new type of CD became available: CD-Recordable, or CD-R. These discs differ from regular CD's in having a light-sensitive organic *dye layer* which can be "burned" to produce a chemical "dark" spot, analogous to an ordinary CD's *pits*, that can be read by existing CD and CD-ROM players. Such CD's are also known as WORM discs, for "Write Once Read Many." A rewritable version *based* on excitable crystals and known as CD-RW was introduced in the mid-1990s. Because both CD-R and CD-RW recorders originally required a computer to operate, they had limited acceptance outside of use as computer software and data backup devices.

To handle the *proliferation* of ever-larger multimedia files (audio, graphic, and video) in computer games, educational software, and electronic encyclopaedias--as well as high-definition movies for television entertainment systems--an expanded storage medium, digital versatile disc (DVD), was introduced in 1995.

DVD

In 1995 Philips and Sony introduced a new type of disc, known as a digital versatile disc (DVD), which was able to store up to 4.7 gigabytes of data, such as high-definition digital video files. A DVD has the same dimensions as a standard CD but cannot be read by a standard CD player, although a DVD player can read standard CD's. DVD players use a higher-power red *laser* (0.65 micrometer) that enables smaller *pits* (0.4 micrometer) and separation *tracks* (0.74 micrometer) to be used.

Figure 12: CD-ROM vs. DVD



The DVD player uses a *laser* that is higher-powered and has a correspondingly finer focus point than that of the CD player. This enables it to resolve shorter *pits* and narrower separation *tracks* and thereby accounts for the DVD's greater storage capacity.

MOD Magneto-Optical Disks

In magneto-optical disks, which can be erased and rewritten, information is written into or read from the disk by means of the magnetic properties of spots on its surface. In reading, spots with different directions of magnetization give different polarization in the reflected light of a low-power *laser* beam. In the writing *process*, which erases all *previous* information, every spot on the disk is heated by a strong

laser beam and is then cooled under a magnetic field. Thus every spot is magnetized in one direction; in other words, every spot stores 0. Then, reversing the direction of the magnetic field, only desired spots are magnetized in the opposite direction by a strong *laser* beam, storing 1.

Magnetic Tape

Tape is one of the more inexpensive *options* for auxiliary memory. Tape access time is slow, since the data must be accessed sequentially, so tape is primarily used for backup copies of a system's memory.

Tape is available in cassette form (common sizes are comparable to the cassettes for a portable tape player and VCR tapes) and on reels (diameter is approximately one foot).

Digital audio tape (DAT) is replacing other forms of tape backup system in newer computer systems.

A single 4-mm-*wide* DAT cartridge, which fits in the palm of your hand, can hold over 2 gigabytes (GB) of data (1 GB = 1024 MB).

Aircraft Application

In an old type of Digital Flight Data Recorder, airplane operating parameters with relation to time are recorded for later use during performance evaluation or during accident investigation. This recorder has a write- and a erase head and stores the last actual flight parameters.

The old Quick Access Recorder (QAR), or also known as Performance Maintenance Recorder (PMR) uses a 'Tape Low' indicators on it. Recording of parameters begins on *track* one of the tape. At the end of the *track* (starting with one), the recorder senses the end-of-tape, changes direction and records on the next *track* in the opposite direction and so on, until the sensing end-of-tape is detected. Because this recorder type has only a write head, the cassette has to be changed before reaching the end of tape.

Computer Self Monitoring Methods

Today, computers are making decisions that are affecting our lives everyday. We are finding that computers are being used to help us cook meals, heat our homes, drive our cars and fly our airplanes. When the computer makes a decision, it must be the RIGHT decision every time. This becomes very important when human life is *at stake*. Even as great as they are, computers or something in their system can become defective, so we must have some way of detecting this when it does *occur*. one of the common methods used is called self-monitoring, self-checking or just simply selfcheck. These methods vary *considerably* from one designer to another.

Self-monitoring is used to help ensure that the computer or its system is working properly and is capable of performing the function for which it was designed. There are provisions for some kind of action upon the *detection* of a failure.

Categories of Self-Monitoring

A self check can be one simple check or a very complex set of checks, *containing* many different types of checks'. This usually depends on many factors, some of which can be cost, application and degree of safety needed.

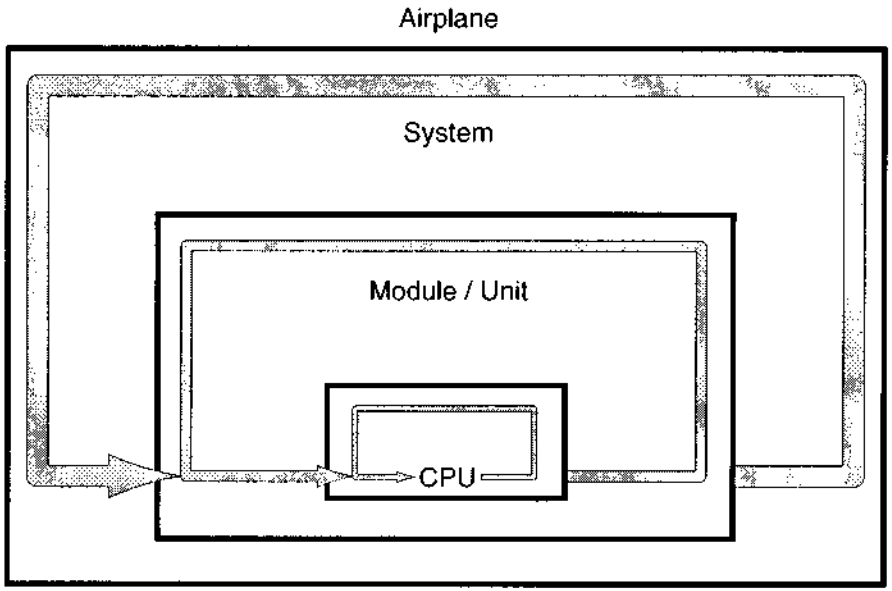
Generally speaking most self-check's are software generated in one form or another. The *Central Processing Unit* will follow a program with the routines necessary to complete all of its self-check's.

There are many categories of self-check's but they will usually be one of the following categories or a combination of one or more.

These categories are:

- CPU circuit (including microcomputers),
- Unit or module circuit or system circuit self-check's.

Figure 13: Categories of Self-Monitoring

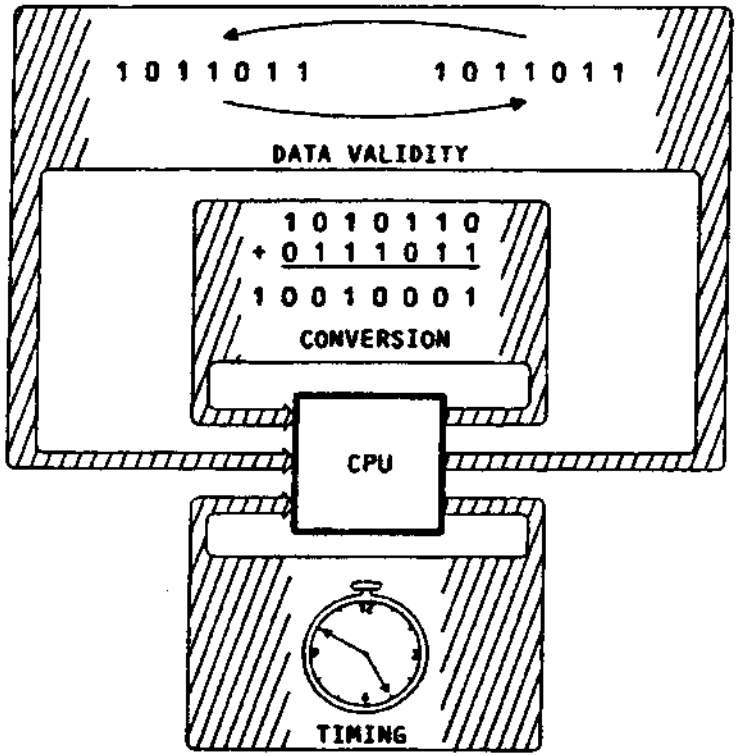


CPU Circuits

This method is one of the more important self- check's, due mainly to the fact that most failures can be detected almost immediately. The faster the problem is detected, the faster the failure action can be taken. The other methods can have some form of *delay* built-in that can *delay* response time. Although all self-check's involve the CPU in one way or another, this method is usually *distinguished* by it being incorporated into one or more of the CPU's STATUS, CONTROL, ADDRESS or DATA BUS lines.

The following are the three most commonly used forms of CPU self-check's. They are conversion, timing and data validity.

Figure 14: CPU Circuits

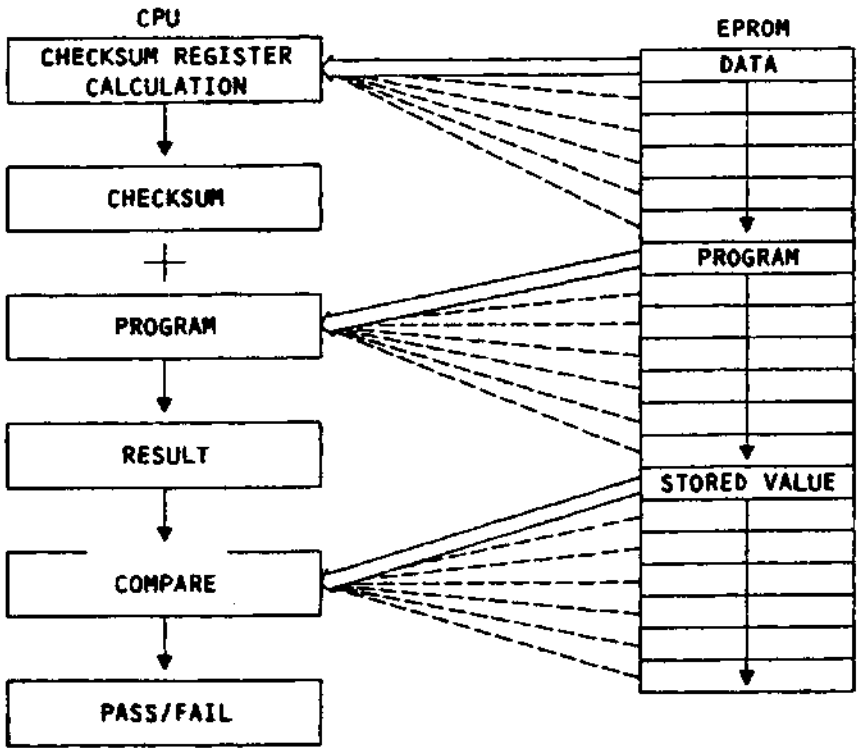


Conversion

This method of self-checking can be used to verify the ROM. There are different ways this can be *accomplished*, depending on the designer.

Some of the names used for checking ROM (usually EPROMS) are checksum, sumcheck or most often just ROM check. This check is used to see if the programming has been changed or altered in some way. We can do this by the checksum method of calculation. An example of this is a data stream from the EPROM which is calculated in the checksum register and this checksum is added to the program. The result is then compared to a stored value to get a pass or fail condition.

Figure 15: Conversion

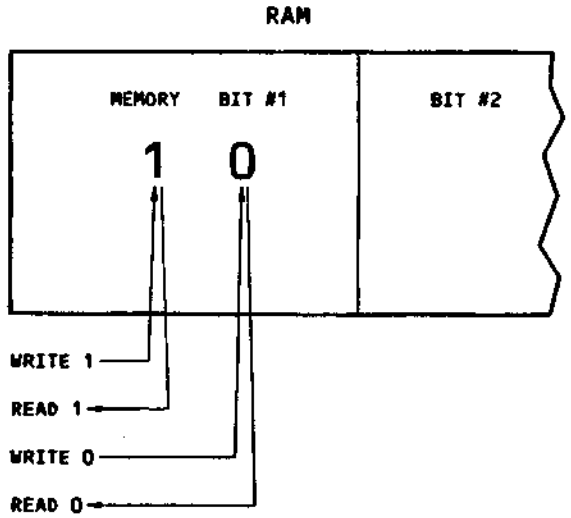


Data Validity

To check the areas of the computer, like the RAM, the registers in the CPU, etc., we can use a method of self checking called data validity. Data in the form of ones and zeros is put into memory and register locations to see if they will "toggle" (change from high to low, or from low to high). An example of a RAM test is to write all ones into all memory location then read them to make sure they are all ones. Next we write the *complement* in all locations, and read if they are all zeroes. The registers can be checked by loading the registers with a "walking" pattern of one/zero and then by reading back the pattern.

There are many other types of checks that can be done with this method of testing. The examples shown are but a few.

Figure 16: Data Validity



Timing

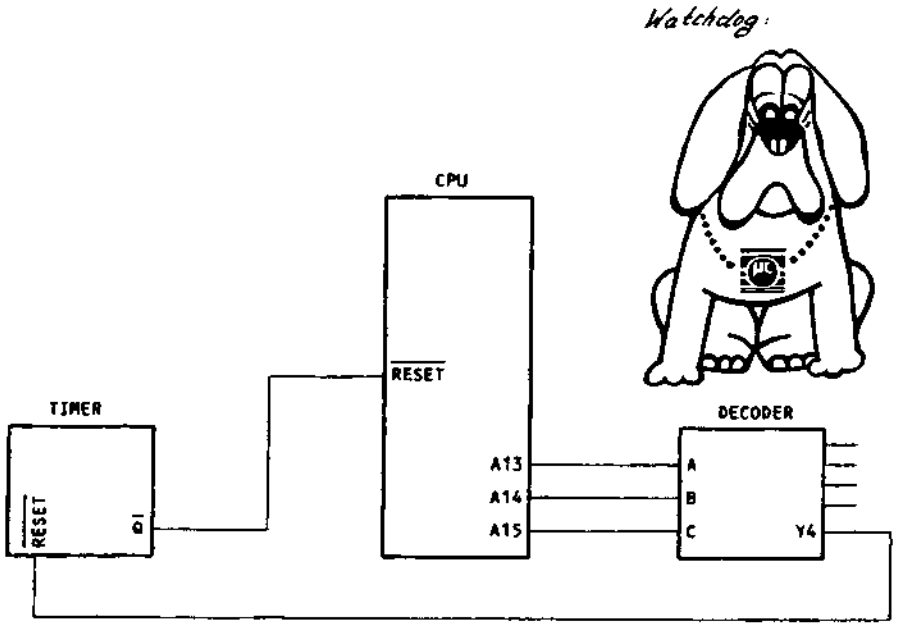
Timing in a electronic circuit is usually a very important *consideration* but it becomes even more important in a computer circuit. Computers must *process* information at a high rate of speed and tasks must be completed at the proper time.

Timing includes the internal timing of the CPU and timing of its support circuitry, like ROM, RAM or anything that *provides* information to the CPU.

The methods of checking computer timing are many, but one of the most common is comparing the timing to a separate timing circuit. This circuit has many names and designs but the most used is the "watchdog timer".

The basic design of a watchdog timer usually uses a running clock that must be stopped or reset before a certain number of clock pulses have occurred. The CPU will have an output to stop or reset the clock. If the clock circuit "times out", the timer circuit will reset or interrupt the CPU to indicate a problem has occurred.

Figure 17: Timing



Module Self-Check

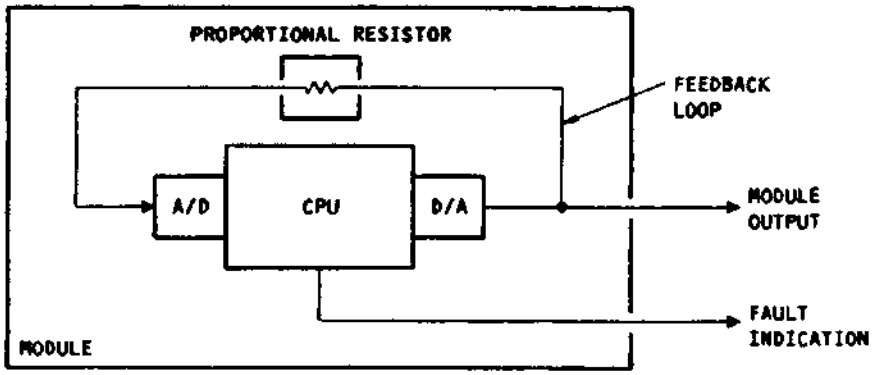
Just as the CPU circuits needed a type of self-check, so does one step above the CPU at the module *level*. The CPU will still be utilized to execute the instruction of the self-checks. *Instead* of checking the components directly *associated* with the CPU, such as the ROM and RAM, the circuits of the module will be checked.

There are many methods of module self-check. Some of the more common ones used are feedback, wrap- around, and comparison. Each will be discussed individually.

Feedback

The most *widely* used form of self-checking is to use feedback. Feedback is the *process* of using a portion of the output fed back to the source to allow monitoring of that output. One example would be a portion of an analog signal fed back through a A/D *converter* and into the CPU for comparison. The program would already *contain* the ratio of the output to the feedback and the monitoring would take place any time there was an output. Figure 18 illustrates the use of feedback self-checking.

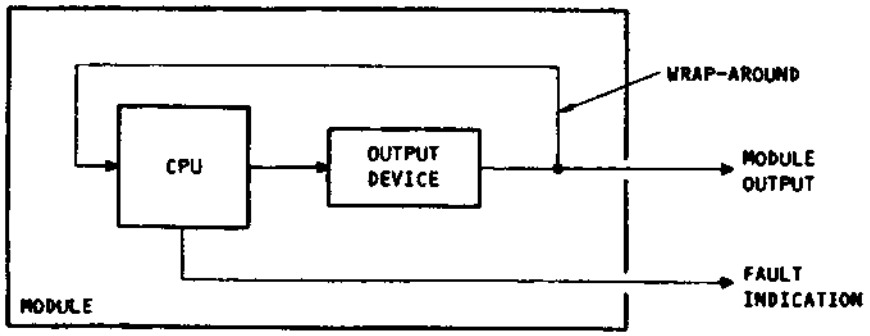
Figure 18: Feedback Self-Check



Wrap-Around

Another form of feedback is called wrap-around. This method usually *employs* the full output being brought back to the CPU for comparison rather than just a portion of it as in the feedback method. Quite often when the data bus is used as the output from the CPU, the output data is latched into registers. The CPU then checks the condition of these registers and compares their contents with the output just sent out. The program dictates what happens if the comparisons don't compare. In another wrap-around method the data bus is used to control an output through a *buffer*. When the computer changes an output, the signal from the output device is wrapped back to the data bus where the computer can monitor the *state* of that particular output. In the case of a digital output being *converted* to an analog one, the analog signal is wrapped back to the input section. There it is *converted* back to digital from analog and compared with the digital output from the CPU. Figure 19 illustrates the use of the wrap-around self-check method.

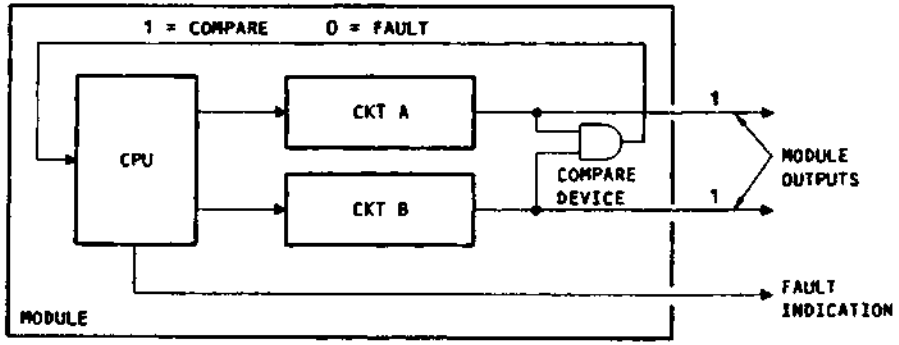
Figure 19: Wrap around Self-Check



Comparison

Certain circuits within a module that performs the same function are compared as a form of self-monitoring. The outputs from both circuits are sent to a comparison device. If one of the outputs doesn't *correspond* to the other, a fault condition is detected and displayed. Figure 20 shows an example of circuit comparison as used for self-checking.

Figure 20: Comparison Self-Check



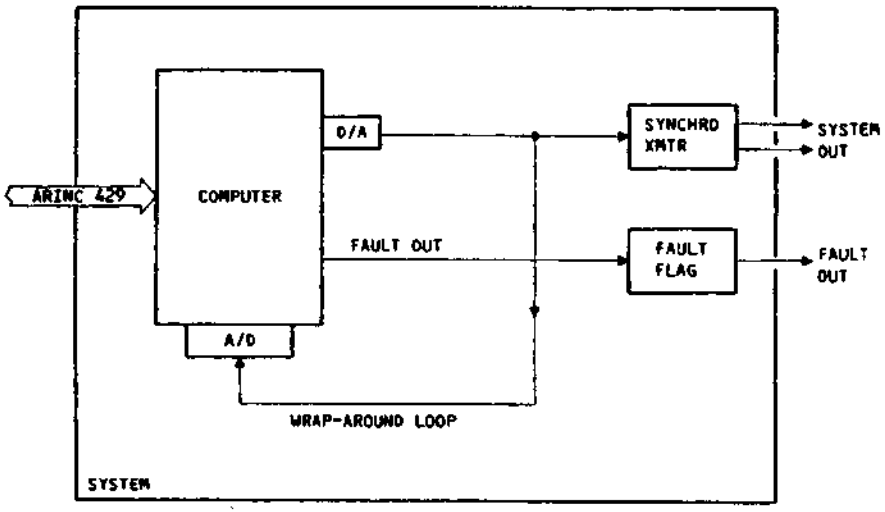
System Self-Check

Self monitoring or checking at a systems *level* usually involves more complex and lengthy check methods. Usually a mechanical action, in combination with an electrical stimulus, is used in self-checking the workings of a system. In an airplane, a systems self-check is usually more expensive to incorporate because of the additional hardware needed to *provide* the feedback to the computer. Some of the hardware involved might be transducers, *discrete* switches and relays, plus all the *associated* wiring and connectors. Because of this additional hardware, and *weight*, it may often be more economical to have the self-test performed at the module *level*.

The methods for self-checking a system are similar to the methods used in self-checking a module. Feedback *loops* are usually the type which allows the computer to check the position of a driven device within the system. Wrap-around may also be used with a system to *convert* the digital data sent out with a wrapped-around input. The comparison method of self check may be used to compare the drive signals to the flaps of the airplane, making sure both were equal.

One example of a system self-monitor is a Digital Analog Adapter which *converts* ARINC 429 information to *discrete* and analog signals. *Heading* is brought in and *converted* from digital to analog information for a synchro transmitter. After the digital to analog conversion, the wrap-around enables the *converted* analog signal to be looped back to the input section of the computer. There, it is *converted* back to digital and compared to the digital output from the computer. If the two are not the same, the fault flag is set.

Figure 21: System Self-Check



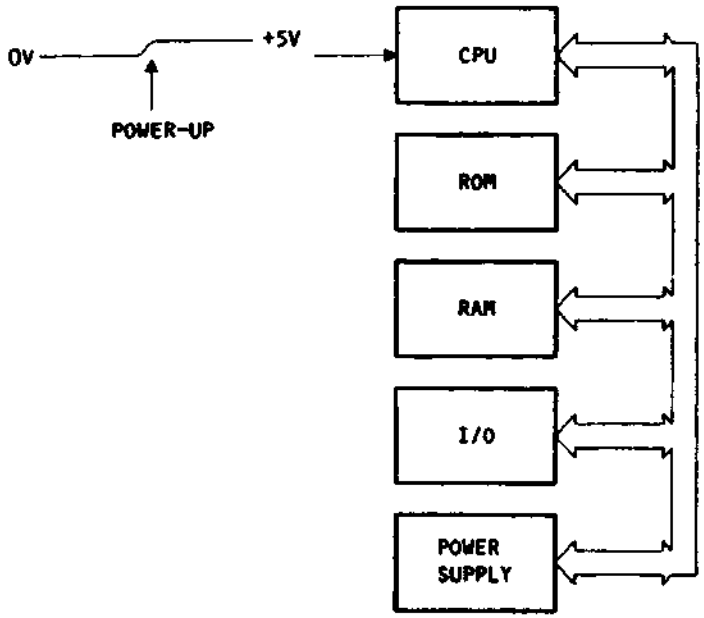
Power-Up

Every circuit, regardless of its complexity, size or function must start from a power down *state* and go to a power-up *state* to perform its task. What better time to check the operation of the circuitry than when it is initially turned on?

A typical system which *employs* a power-up self- check may be a yaw damper system. Once the power is applied, the CPU performs tests on the *associated* circuitry, such as ROM, RAM and timing circuitry. Input conditions may be set to certain *states* so the CPU can check digital and analog signals, and/or *discretes*.

Devices connected to the outputs such as actuators may also be driven to certain positions to ensure they will move when directed in the execution of the main program. Other *discrete* outputs, such as lights and indicators may also be reset. Power supplies and their reset signals are checked. Once the complete power-up self check is completed and all passes, the main program is entered. If there are failures, the power-up self check may restart the system and record the failure in the maintenance memory.

Figure 22: Power Up Test

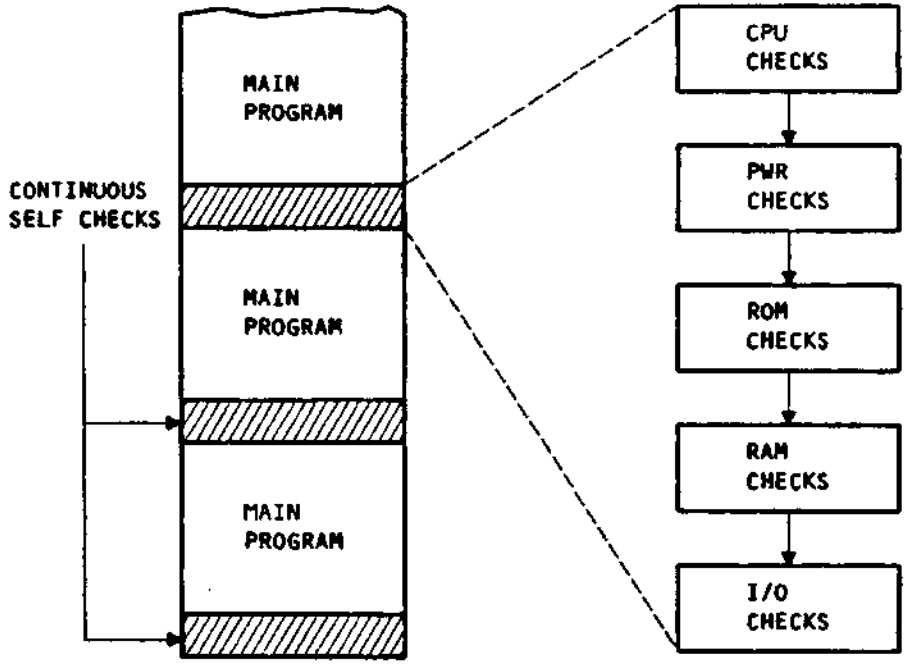


Continuous Self-Check

Monitoring during the execution of the main program of a system can be termed "continuous". The urgency of the system usually dictates whether this type of self checking should be used or not. If conditions aboard the airplane exist in which the failure of that system would be a safety factor, continuous monitoring is used to constantly check the system performance. Failure of this continuously monitored system usually results in a *redundant* system coming on-line and the first system is set to a power down condition.

Continuous self checks perform partly the same self tests as at power-up. Usually the RAM and ROM are checked, as well as the power supply monitor and input/ output circuitry. Continuous self checks are usually much shorter than the power-up checks because of the large amount of *processing* which must take place between the self-checks.

Figure 23: Continuous Self-Check



Types of Computer Self-Monitoring

Computers used in the airplane industry today may *contain* one, two, or more CPU circuits. Each may be *processing* the same, similar or completely different information all within the same computer.

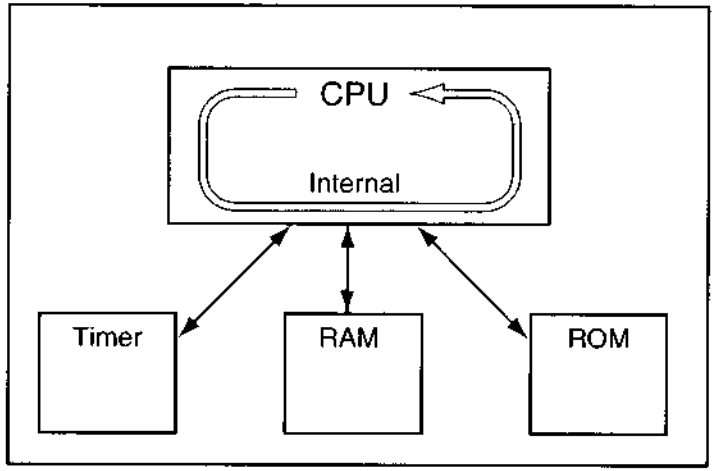
We can have a computer that *contains* one CPU monitoring itself but if there are two or more, they can check themselves or each other. Within a computer we can have one or more of these types of computer self-checking.

The three types are single, interacting and monitor circuit.

Single

When a computer *contains* only one CPU we are limited to the CPU self-checks discussed elsewhere in this section. This will usually include ROM check, RAM check, timing check and internal CPU check. Other checks may be included but will vary with the design *requirements*.

Figure 24: Single Computer Self-Monitoring

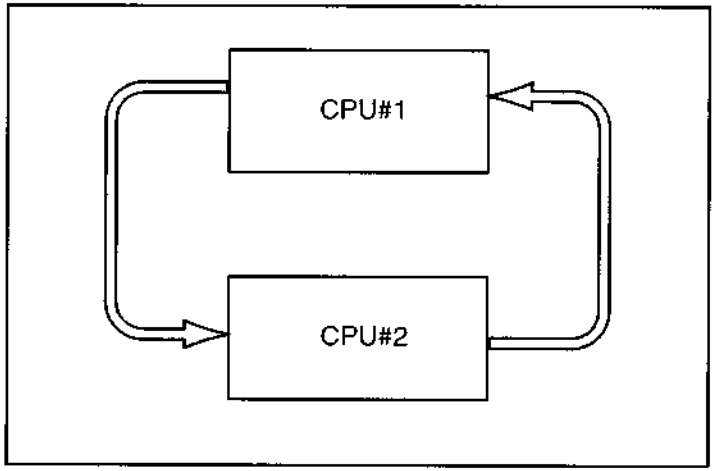


Interactive

The computer that *contains* more than one CPU may be *processing* different information and for all practical purposes can be *considered* single CPU's *contained* in one housing. The ones that are *processing* the same or very similar information can use interactive checks, in addition to the normal CPU self-check's.

When two or more CPU's check each other, this is called interactive checking. These can take the form of "Handshaking", "Watchdog timing", "information exchange" to name a few. The complexity of these checks depends on the design *requirements*.

Figure 25: Interactive Computer Self-Monitoring

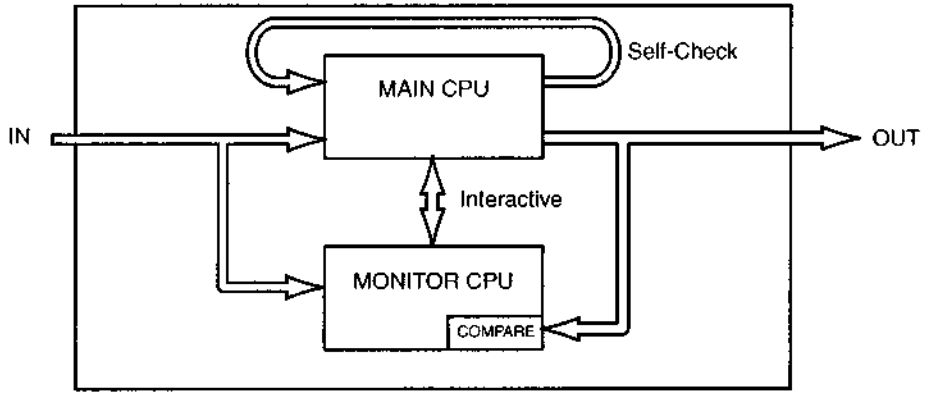


Monitor

A computer can have a separate CPU installed for the sole purpose of checking that the CPU is performing its designed function. The monitor CPU can be connected to one or more *processing* CPU's even if they are *processing* different information. The operation of the CPU and MONITOR CPU is similar to the interactive self-checking.

The monitor checking offers another way of self-checking in that it can *process* the same information as the main CPU and then compare its *processed* information to the main CPU's *processed* information for *accuracy.* This type of checking is usually expensive and used most often in critical systems.

Figure 26: Monitor



Computer Technology as applied in Aircraft Systems

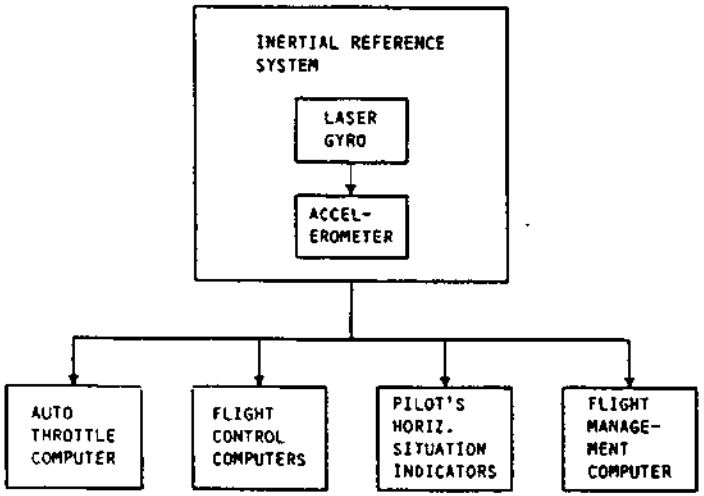
Reference Computer

Although computers can be classified by hardware as analog, digital, or *hybrid*, they are more often classified by their tasks or application. A computer which may be used for a source of information or data can be called a reference computer. Reference signals from this computer may be self-*contained* and only *provides* outputs.

An *Inertial Reference System* (IRS) is one example of a reference computer. This system is a *laser* gyro and accelerometer *based* reference system used to generate such outputs as airplane *attitude*, *heading*, acceleration and *angular* information. Other than for initialization purposes, the IRS needs no inputs to perform its task.

Some of the units utilizing this information as a reference are the Flight Control Computers, the pilot's Horizontal Situation Indicators, and the Flight Management Computer.

Figure 27: Reference Computer



Informational, Warning and Display Computers

A computer system that collects data from various places, *processes* it, and formats it for display and warning can be called an Electronic Instrument System (Airbus) or an Engine Indication and Crew *Alerting* System (Boeing).

The main task of such a computer system is to collect data and display it in a central place. During the different phases of a flight, from power up through touchdown, the flight crew is often in need of information concerning a certain airplane system.

Information needed may include for example total air temperature, engine oil *levels*, hydraulic pressures, and engine vibration *levels*. On the ground, the maintenance personnel often need to recall certain events that occurred during the flight, such as out of normal parameters on an engine (overspeed), or Auxiliary Power Unit voltage information.

The flight crew has various types of information available to them before, during and after a flight. Parameters used to set and monitor engine thrust are displayed on a cathode ray tube (CRT) full time and the remaining engine parameters may be selected for display by the crew.

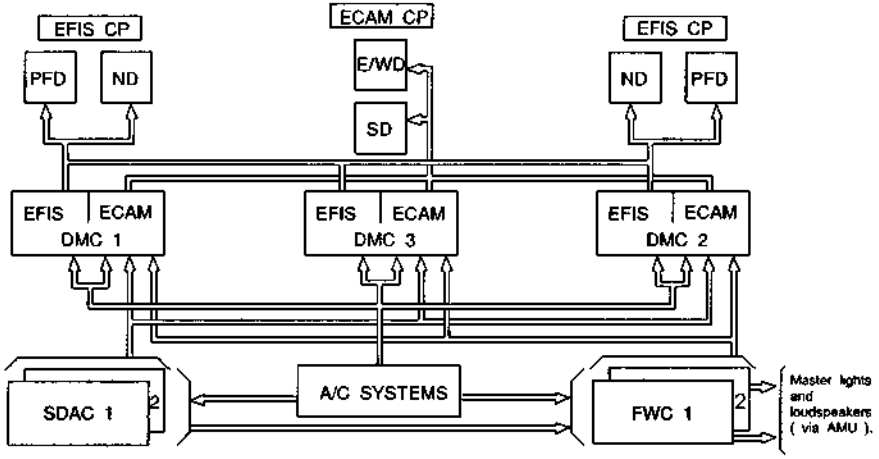
Maintenance information can be displayed when required by maintenance personnel. Airplane configuration, equipment cooling and status, electrical/ hydraulic parameters, performance data and engine exceedance are some of the types of maintenance information available.

On Figure 28 the Airbus Electronic Instrument System is shown. It is divided in an Electronic Flight Instrument System **EFIS**, and an Electronic Centralized Aircraft Monitoring **ECAM**. The EFIS *provides* flight information and the ECAM *provides* system and warning information. The Electronic Instrument System *comprises* seven computers:

- three identical Display Management Computers (DMC's)
- two identical Flight Warning Computers (FWC's)
- two identical System Data *Acquisition* Concentrators (SDAC's).

The DMC's *comprise* two independent parts: one for the EFIS function and one for the ECAM function.

Figure 28: Example of the Airbus Electronic Instrument System



Controlling Computers

A computer with the primary task of controlling something can be called a controlling computer or controller. This is one of the largest categories of computers.

In industry today nearly anything that can be controlled by a computer. Computerized controllers range from simple temperature controllers to entire systems for controlling a complete factory.

Airplanes have a variety of systems, surfaces, and devices needing control during operation, both in the air and on the ground. It is impractical to have the flight crew manually control all of the necessary systems, so computers are used to lighten the crew's workload by providing automatic control.

An example would be the control of the slats and the flaps on an Airbus A330:

The Slat Flap Control Computer (SFCC) *provides* a means to monitor the slat and flap lever position and to control the slat and flap position on the wings.

Position of the surfaces is selected from a control lever which transmits the demand to the Slat Flap Control Computers through a Command Sensor Unit.

The SFCC's, which are identical, ensure the control and monitoring of the slat and flap system. On lever demand, the SFCC's send signals to a Power Control Units to energize the valve blocks.

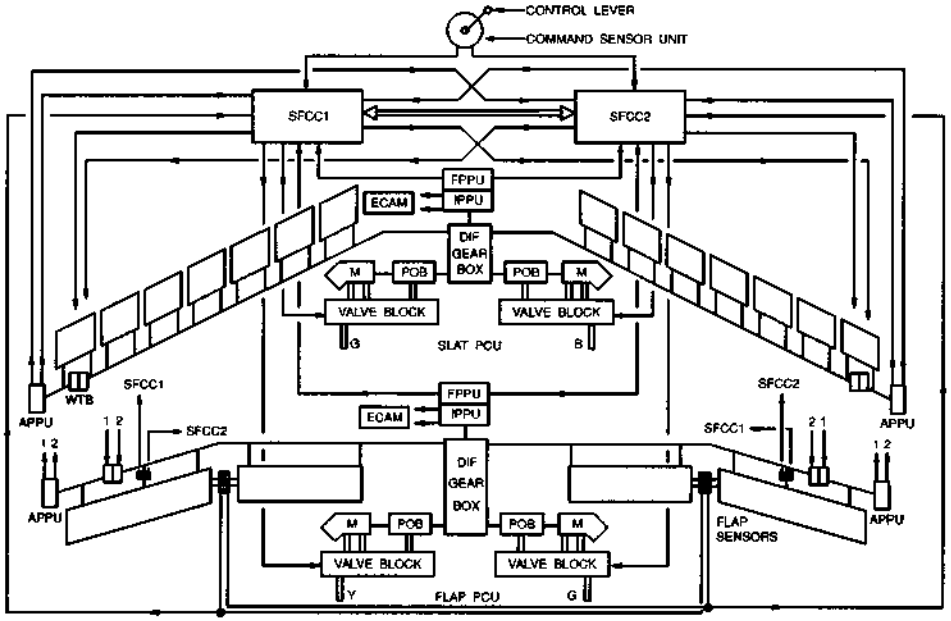
Two hydraulic motors in each Power Control Unit *provide* hydraulic actuation. Each of them is powered by a different hydraulic system and has its own valve block and pressure off brake. The valve blocks control the direction of rotation and the speed of their related PCU output shaft.

The hydraulic motors move the transmission through a differential gearbox. Then torque shafts and gearboxes transmit the mechanical power to the actuators which drive the surfaces.

Wing tip brakes are *provided* in order to stop and lock the system when certain types of failure are detected.

Feedback signals sent by dedicated Position Pick-off Units are used by the computers to control and monitor the system. A Feedback PPU *provides* signals concerning the PCU output shaft position, while two Asymmetry PPU's send information about surface actual position. In addition signals sent from an Instrumentation PPU are used for position indicating on the ECAM.

Figure 29: Slat Flap Control System



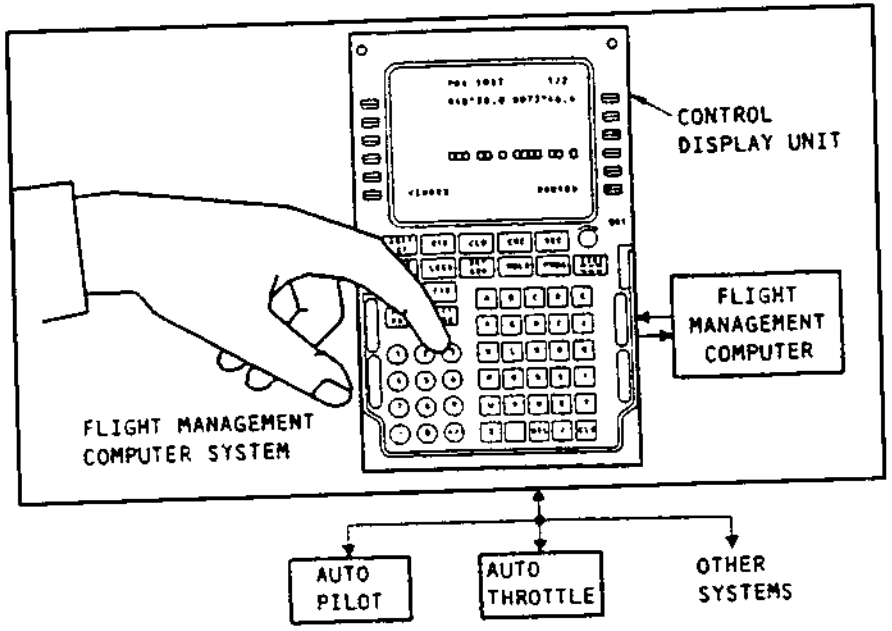
Interactive Computers

Some computers can perform several different tasks depending on operator inputs. Such a computer is called an interactive computer. Interactive computers typically display information to the operator and then manipulate the data *based* on the interaction between the operator and the computer.

The Flight Management Computer (FMC) is an example of an interactive computer. The flight crew interfaces with the computer by means of the Control Display Unit (CDU) to input performance data, initialization data and route structure.

The computer calculates optimum cost profiles for climb, cruise and descent used by the autopilot and auto throttle for automatic flight control. All computed values are also automatically displayed allowing the crew to fly an optimum profile using manual control.

Figure 30: Interactive Computers



Airplane Digital Systems - Summary

Airplanes typically have many computers to control, monitor, *provide* references, and make available information. These computers can be either analog, digital or *hybrid*. It can also be noted that computers are typically different combinations of the types of computers as categorized by application.

Those are interactive, reference, information, warning, display, and controlling, computers.

As in the example of the interactive computer, the flight management computer is also used as a control computer by controlling the autopilot and other systems. It can be an Informational computer by providing the flight crew with route Information.

The flight management computer also acts as an information computer by storing information to be used by other airplane systems. It also acts as a warning computer, In that it monitors many of its inputs and stores this information for further reference. Computer types and applications are as varied as their tasks.

5.10 Fibre Optics

Introduction

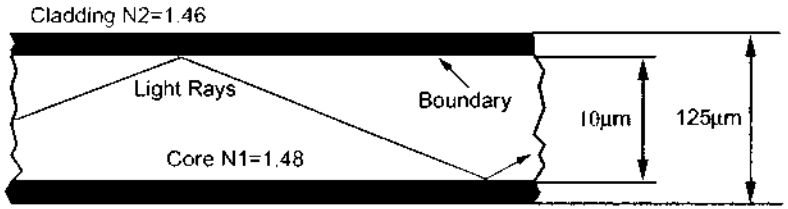
Fiber Optics is the science of transmitting data, voice, and images by the passage of light through thin, transparent fibres. In telecommunications, fibre optic technology has virtually replaced copper wire in long-distance telephone lines, and it is used to link computers within local area networks. Fibre optics is also the basis of the fibrescopes used in examining internal parts of the body (*endoscopy*) or inspecting the interiors of manufactured structural products.

Optical fibres are glass or plastic *waveguides* for transmitting visible or infrared signals. Since plastic fibres have high *attenuation* and are used only in limited applications, they will not be *considered* here. Glass fibres are frequently thinner than human hair and are generally used with LED's or *semiconductor lasers* that *emit* in the infrared region. For *wavelengths* near 0.8 to 0.9 μm, gallium arsenide-aluminium gallium arsenide sources are used, and, for those of 1.3 and 1.55 μm, indium phosphide-gallium indium arsenide phosphide sources are *employed*. As noted earlier, optical fibres *consist* of a glass core region that is surrounded by glass *cladding*. The core region has a larger *refractive* index than the *cladding*, so that the light is confined to that region as it *propagates* along the fibre.

A typical glass optical fibre has a diameter of 125 micrometers (μm), or 0.125 mm (0.005 inch). This is actually the diameter of the *cladding*, or outer reflecting *layer*; the core, or inner transmitting cylinder, may have a diameter as small as 10 μm (see Figure 1).

These are typical values, Fibre core diameters can range between 1 and 100 μm, while *cladding* diameters can be between 100 and 300 μm.

Figure 1: Basic Construction of glass optic fiber



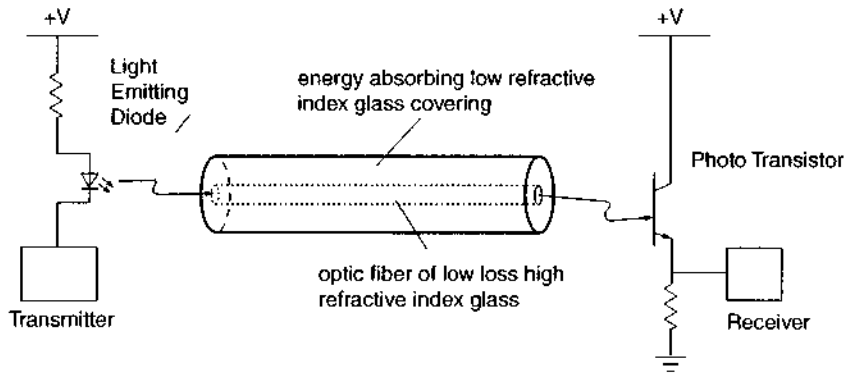
Fiber optic transmission principle

Through a *process* known as total internal reflection, light *rays* beamed into the fibre can *propagate* within the core for great distances with remarkably little *attenuation*, or reduction in intensity. The degree of *attenuation* over distance varies according to the *wavelength* of the light and to the composition of the fibre. When glass fibres of core/*cladding* design were introduced in the early 1950s, the presence of *impurities* restricted their *employment* to the short lengths sufficient for *endoscopy*. In 1966, electrical engineers K.C. Kao and G.A. Hockham, working in England, suggested using fibres for telecommunication, and within two decades silica glass fibres were being produced with sufficient *purity* that infrared light signals could travel through them for 100 km (60 miles) or more without having to be boosted by repeaters. Plastic fibres, usually made of polymethylmethacrylate, polystyrene, or polycarbonate, are cheaper to produce and more flexible than glass fibres, but their greater *attenuation* of light restricts their use to much shorter links within buildings or automobiles.

As mentioned earlier, optical telecommunication is usually *conducted* with infrared light in the *wavelength* ranges of 0.8-0.9 μm or 1.3-1.6 μm *wavelengths* that are efficiently generated by light-*emitting* diodes or *semiconductor lasers* and that suffer least *attenuation* in glass fibres. (see Figure 2). Fibrescope inspection in *endoscopy* or industry is *conducted* in the visible *wavelengths*, one bundle of fibres being used to illuminate the examined area with light and another bundle serving as an elongated lens for transmitting the image to the human eye or a video camera.

Fibres with a larger core diameter are called multimode fibres, because more than one electromagnetic-field configuration can *propagate* through such a fibre. A single-mode fibre has a small core diameter, and the difference in *refractive* index between the core and *cladding* is smaller than for the multimode fibre. Only one electromagnetic-field configuration *propagates* through a single-mode fibre. Such fibres have the lowest *losses* and are the most *widely* used, because they *permit* longer transmission distances.

Figure 2: Light Transmission Principle



Advantages of Fiber Optic Data Transmission

The advantages of light energy along a glass *fiber* for onboard communications are great. Freedom from signal *egress* and *ingress* is *beyond* that of the most sophisticated wired system.

Electromagnetic fields at frequencies below that of light will not interact with glass. As long as the *fiber* is shielded from light, easily *accomplished* even for intense light, the *fiber* is isolated from external radiated energy.

Wide bandwidth is another important characteristic of *fiber* optics. *Bandwidths* of gigabits per second are possible. It may appear this is more *bandwidth* than could ever be used in an aircraft, but EW (electronic warfare) equipment for the military has large *requirements*.

Civil aircraft with graphics displays can also use gigabit per second rates. For most applications, gigabits per second is more than enough.

Fiber optic systems have low *losses*. This is not critical for aircraft because distances between terminals are short. *Losses* are extremely important for telecommunications where *losses* translate to additional repeaters and high cost.

Fiber optics are lighter in *weight* than wired systems. *Weight* per unit length of *fiber* is less than the *equivalent* length of a twisted pair. It would seem that a small *strand* of glass is significantly lighter. However, when the *fiber* is encased in protective and strength members, the *weight* advantage is reduced.

Consider that *fiber* carries ten times the data of twisted pair, and compare *weight* per unit length to ten twisted pairs, the *weight* advantage of *fiber* becomes significant. To compare *fiber* to wired communications, therefore, the comparison should be kg per meter per bit per second, using the proper SI units.

The most important advantage of *fiber* for aircraft is immunity to high intensity radiated fields or HIRF (often from high power transmitters on the ground). This is particularly true for critical applications such as fly by wire, optics, or "fly by light".

Disadvantages of Fiber Optic Data Transmission

Fiber optics are not without disadvantages. First, it is difficult to work with. If a copper wire breaks, it is repaired with a crimp connector or soldering. As long as wires touch, and electrical connection is made.

It's not that simple with *fiber* optics. Connectors are also a problem. A connector for copper wires is simply two *conducting* pieces that *provide* physical contact and a circuit. Connecting two *fibers* requires microscopic *fibers* to be perfectly aligned and joined so light energy couples from one *fiber* to the other.

The *fiber* is fragile. It can be protected by encasing it in a protective housing but the *fiber* can still be fractured. A common *fiber* damage is related to bending. If the *fiber* is bent in a tight radius, *losses* increase and there is risk of fracture. Mishandling causes fractures by pulling the *fiber* tightly over a sharp corner or stepping on it. This is dangerous and should be avoided for either technology. For wire, sharp bends do not usually cause immediate failure and, if the wire is straightened before installation is complete, no permanent damage may result. These problems are solved by training and procedures required of any new technology.

Fiber optics are the perfect solution to a communications system that needs immunity to HIRF. In spite of the disadvantages, more *fiber* will be applied in aircraft.

The Boeing 777 has a communication network that uses *fiber* optics.

Construction of the Cable

Each fibre optic *strand* is 0.0055 inches in diameter this is covered with several *layers* of material, *cladding* (to keep light in) primary *buffer* (protects glass fibre during manufacture), secondary *buffers* are coloured to *permit* identification of each fibre optic *strand*. The *strand* is now 0.035 inches in diameter.

Figure 3: Fiber Optic Cable

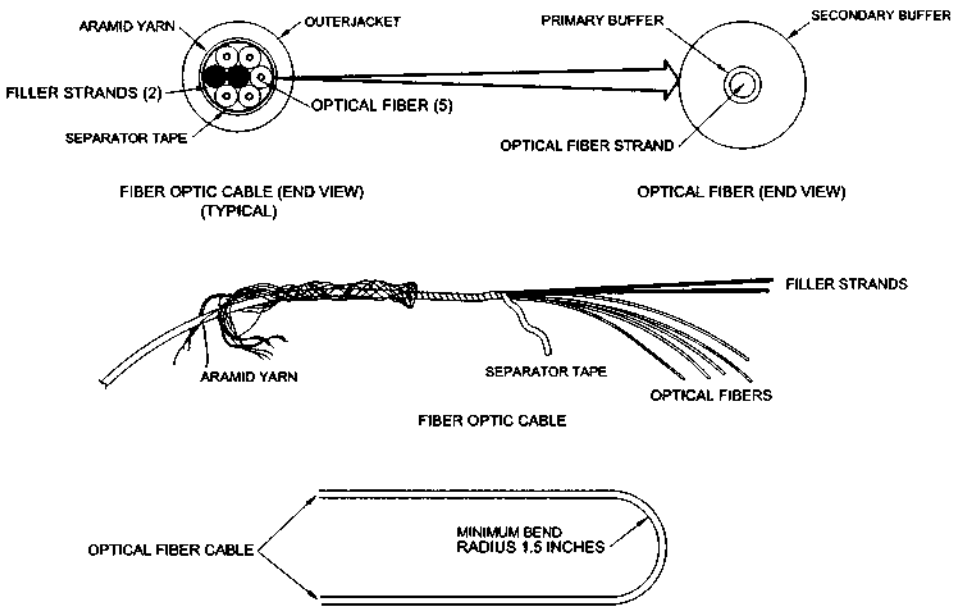
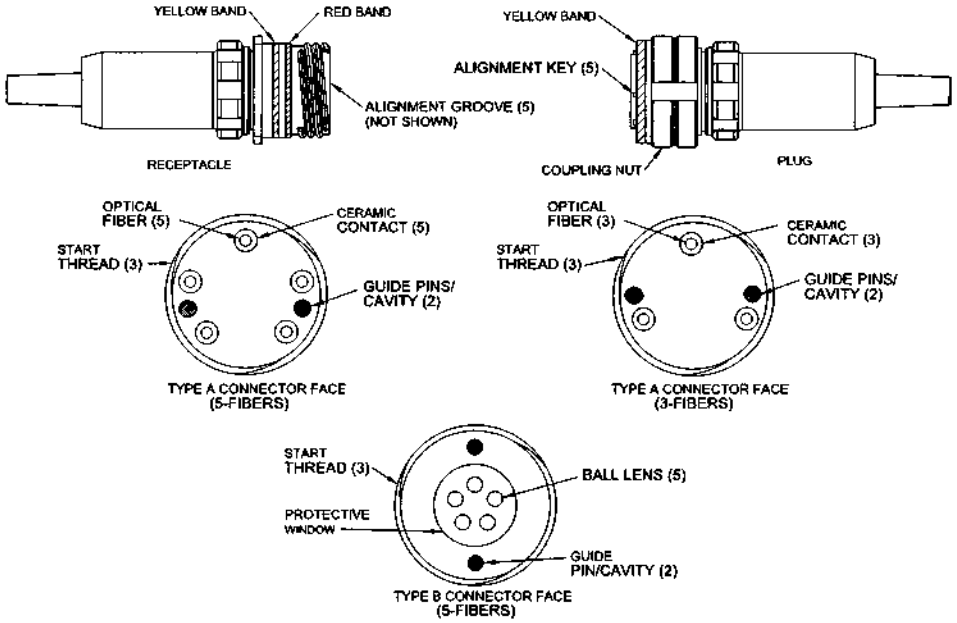


Figure 4: Optical Connectors



The cable itself can have three or five fibre optic *strands*, a number of black filler *strands* may be used to make up the cable these are also 0.035 inches in diameter. A polyester tape covers the *strands*, it makes the cable more flexible. A woven aramid *yarn* protects the optical fibres from damage. The outer jacket is a purple thermoplastic.

Connectors

Type A Connector

Type A -these are used at production breaks that are not regularly connected and disconnected. This is a multi-channel, in line (butt type) connector which has very low light *loss* between the fibre optic components.

The connector has alignment keys on the plug and alignment grooves on the receptacle to *accurately* align the optical components, guide pins in the plug fit into cavities in the receptacle; these pins touch the bottom of the cavities so you cannot overtighten the connector.

The coupling nut on the plug barrel has a yellow band, the receptacle barrel has a red and yellow band. When the red band on the receptacle is at least 50% covered by the coupling nut the connection is correct. Three start threads on the plug make sure of a straight start on first joining. The plug and receptacle have ceramic contacts that touch when connected. The light signal goes through the holes in the end of the ceramic contacts when they are in direct physical contact with each other.

Type B Connector

This type of connector is used to connect to LRU's and is therefore more frequently connected and disconnected. It is a multi-channel, expanded beam (ball lens) con-

nector, light *loss* is low but not as good as the Type A connector. It has the similar alignment keys and guide pins as the Type A.

The connector has a miniature ball lens at the end of each fibre behind a protective window. This lens, expands and focuses the light through the protective windows of the plug and receptacle to another ball lens which narrow the light and sends it into a fibre.

Important maintenance aspects with these connectors are:

1. Before examining, ensure equipment is switch off, light from the optical fibre could damage your eyes.
2. Only use approved procedures to clean the connectors and lenses.

Applications in Aircraft Systems

The Boeing 777 uses a fibre optic communications network called OLAN (Onboard Local Area Network) this is divided into two parts:

1. Avionics local area network (LAN)
2. Cabin LAN

The Avionics LAN connects the following LRU's:

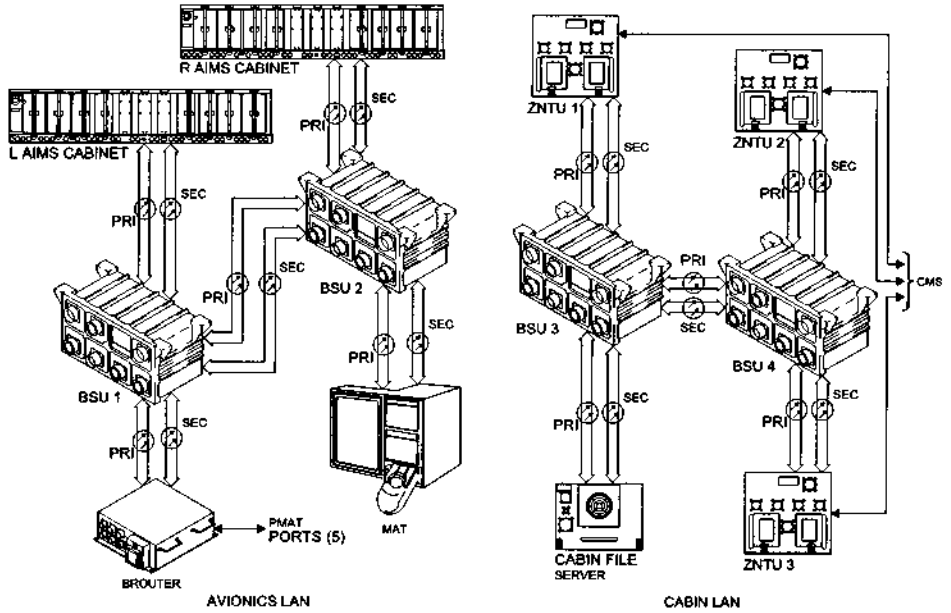
a) Left and right AIMS (Aircraft Information Management System).
b) MAT (Maintenance Access Terminal).
c) First Officers Side Display.
d) Captains Side Display.
e) B-Router.

The B-Router receives and sends signals to LRU's and connects signals to the P MATS (portable maintenance access terminals).

The cabin LAN connects the following LRU's:

a) ZNTU 1,2,3 (Zone Network Controller/Telephone Distribution Network).
b) CFS (Cabin File Server)

Figure 5: Onboard Local Area Network



ARINC 636 is a fibre optic data bus, one *strand* of fibre optic per bus. The two *strands provide* a primary (PRI) and secondary (SEC) bus.

5.11 Electronic Displays

Introduction

Electronic Displays for Aircrafts may be constructed in several ways.

The first instruments for aircraft borrowed technology from ships, railroad locomotives and auto-mobiles - transportation of the period. For simple parameters, dials and gauges were *adequate*. Even the first radio navigation instruments used simple pointer indicators for the ADF and CDI.

In spite of the complexity, many aircraft are still full of mechanical displays.

The ultimate navigation instrument, however, would be a map with the familiar "You are here" arrow. On the display is your flight plan, destination, location of bad weather, obstructions and collision hazards. This display depends on the ability to *pictorially* present information. Other data can be displayed with numbers and letters, or alphanumeric. Examples are engine parameters, radio frequencies, airports, outside air temperature, *airspeed* and so on. Any parameter that is scalar can be shown with an alphanumeric display, while data that are vectors, such as courses, tracks, and *headings*, are best indicated with a graphic display.

A reflective display (like a painted number) requires ambient light to be visible; it does not generate light. All mechanical pointer displays are reflective. For alphanumeric, numbers are painted on rotating drums mounted behind a window on a panel. The drum rotates and displays characters similar to the way a mechanical *odometer* operates in an automobile. The first DME's were made this way; with distance displayed on rotating drums.

An early emissive display was the warning lamp. An example is a gear up - gear down light and a marker beacon indicator. One *requirement* of a warning light is high brightness; it must be seen in sunlight, which can be very bright in a cockpit. There are no trees or clouds to shade the sun from an aircraft at high *altitude*. To reduce direct sunlight, a glare shield creates a "roof' at the top of the instrument panel. Even with this shield, emissive displays must be very bright.

At the other end of the emissive brightness scale is the dark of night, where displays are reduced in intensity. There could also be a single warning light with so much illumination that night vision of the crew would be affected. (This is more critical in military flying than in the civil world.)

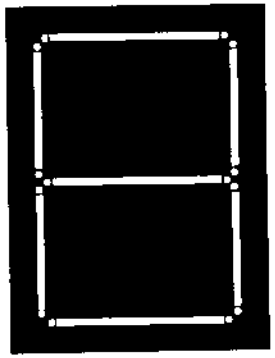
The required range of light intensivity is large and some display types cannot *provide* it.

Older Display Technologies

Incandescent

One of the first technologies in aircraft was the *incandescent* lamp, leading to the first alphanumeric electronic displays. The *incandescent* was probably the first application of the segmented numeric display, where digits are formed by seven segments. Illumination comes from filaments stretched between posts (see Figure 1).

Figure 1: Seven Segment Incandescent Display



The display *provides* 10 decimal digits. Light *emitted* by the display is close to white. When intensity is reduced, the colour shifts to longer *wavelengths* and appears more orange. This was not a bad characteristic because redder light does not affect night vision as much as shorter *wavelengths*. Filters over a display can generate colours other than white. This is a problem, however, when the display is reduced in intensity and shifts toward red.

Although the *incandescent* is still found in many aircraft, there are limitations. First, it is not suitable for alphanumeric displays. There are alphanumeric displays with 16 segments and numerals are *acceptable*. Letters of the alphabet, however, are difficult to read. Another problem is efficiency. At daytime brightness, the display becomes hot, limiting its life to approximately 5000 hours at moderate brightness.

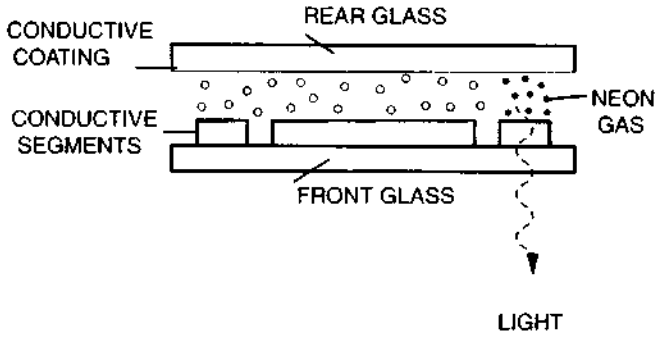
However, when the display operates at sunlight-readable intensity, life is reduced *considerably*. The final problem is cost. When *incandescent* displays became available, the high cost was *acceptable* since they were the only choice. Later technologies are much less costly.

Plasma

A display technology used in significant numbers before LED's was introduced, is plasma. It is based on orange light *emitted* from ionized neon gas. An example is shown in Figure 2. Neon gas is *contained* between two flat glass plates. The rear glass is *coated* with a *conductive* material, usually a metal.

Deposited on the inside of the front glass are metal segments. These can be a seven segment configuration or a customized pattern. The metal segments are very thin and transparent to light. If a sufficiently high electric potential is applied between the rear *conductive* coating and a metal segment, the neon gas ionizes. Orange light is *emitted* when neon atoms return to their zero energy *state*. By selectively energizing segments, numeric digits are generated.

Figure 2: Plasma Display



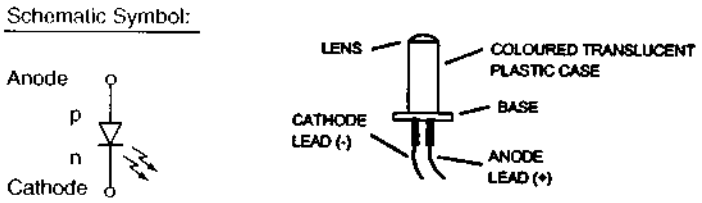
Recent Display Technologies

Light Emitting Diode (LED's)

A LED *consists* of a junction diode made from the *semiconductor compound* gallium arsenide phosphide. It *emits* light when forward *biased*, the colour of the light *emitted* is in direct proportion to the current flow.

Light emission in the red, orange, green and yellow regions of the spectrum is obtained depending on the composition and *impurity* content of the *compound*.

Figure 3: Light Emitting Diode



When a P-N junction is forward *biased*, electrons move across the junction from the n-type side to the p-type side where they recombine with holes near the junction. The same occurs with holes going across the junction from the p-type side. Every recombination results in the release of a certain amount of energy, causing, in most *semiconductors*, a temperature rise. In gallium arsenide phosphide some of the energy is *emitted* as light gets out of the LED because the junction is formed very close to the surface of the material.

In *applying* this to aircraft displays either the 7 segment or *dot*-matrix configurations may be used.

Various seven-segment decoders are available to drive common-cathode and common-anode seven-segment displays. These drivers receive a number, usually in BCD format, and decode the number into signal *levels* to activate the proper a-g segments of the display. Figure 4 shows one example of a seven segment display. Figure 5 shows a TTL 7447 *IC* and a common-anode LED display. The TTL 7448 is designed to drive common-cathode displays. The dc illumination method shown is the easiest to *implement*; but higher light output with lower energy consumption can be obtained by pulsing and multiplexing the display voltage. A pulse rate of 100 Hz is *imperceptible* to the human eye.

Figure 4: Seven Segment Display

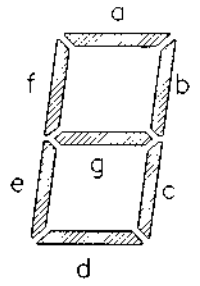
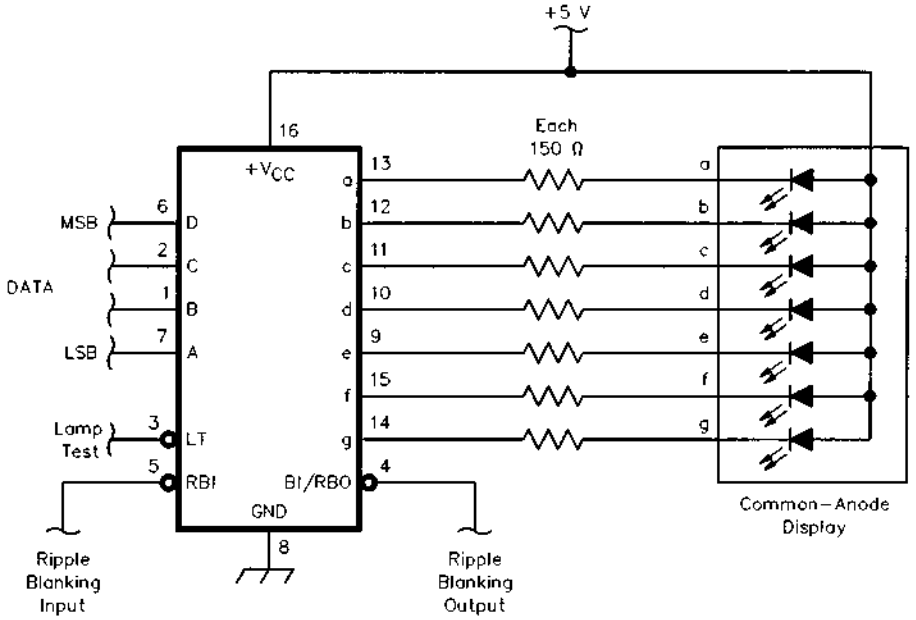


Figure 5: Seven Segment Decoder and Display



Video Displays

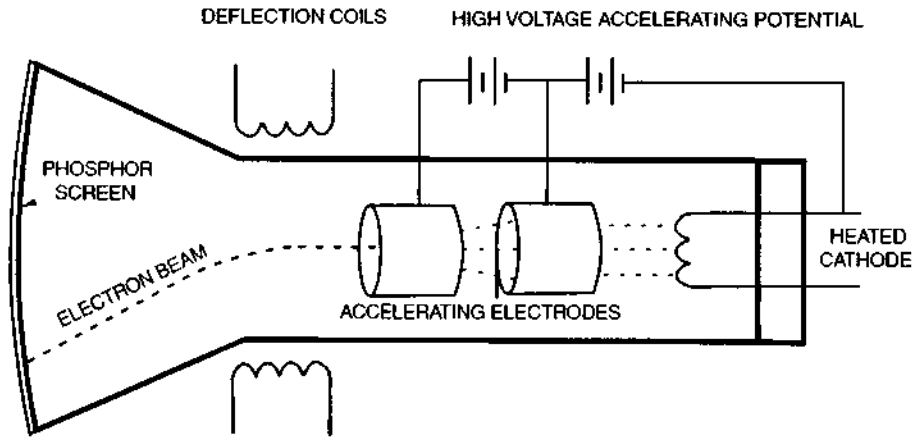
Cathode Ray Tubes (CRT)

Although Active Matrix *Liquid Crystal Displays* (see next section) dominate the cockpits of new aircraft, the CRT has long been one of the most popular technologies. It has been manufactured for more than 75 years and installed in aircraft since World War II. The first application as an airborne display was military radar screens.

In the mid fifties, it was applied to *weather radar* for airliners. When colour was added, different shades of colour, rather than a gray scale, could designate the intensity of rain. *Weather radar* screens were also *enhanced* with overlays and manufacturers started to use the CRT as a multi function display, where check lists and other items are displayed. The *weather radar* was the first true electronic flight instrument system.

Construction of the CRT is shown in Figure 6. It is a vacuum tube with components enclosed in an evacuated glass bottle. At the end away from the screen, in the neck, is a heated cathode to *provide* a source of electrons. The electrons accelerate toward the screen with increasingly higher positive voltage. Electrodes near the cathode focus the beam into a narrow spot on the screen. The final accelerating electrode has a very high potential, typically from 10 to 25 kV, depending on CRT size.

Figure 6: Principle of a Cathode Ray Tube



The electron beam is deflected by a magnetic field. An electric field could be used, but a magnetic field deflects the beam through a greater angle. This *permits* the design of shorter tubes, but even these tubes often have a length that is objectionable for aircraft. A magnetic field is generated with a ferrite *yoke* that fits around the neck of the CRT.

The screen is *coated* with a phosphor material. Single colour (monochromatic) tubes have one phosphor deposited on the front glass. A thin coating of aluminium is evaporated onto the glass to *provide* a return path for current of the electron beam. When an electron strikes phosphor atoms they *emit* light photons. Green is a common colour for CRT's in oscilloscopes and computer screens and is still used in aircraft instruments.

CRT phosphors become sunlight readable by increasing beam current and forcing more photons from the phosphor. The disadvantage is a reduction in tube life. A CRT for a television receiver in the home, where ambient light is low, can last tens of thousands of hours. A CRT in an aircraft will last thousands of hours.

A reduction of beam current decreases CRT brightness without a colour shift and is an effective method of dimming the display.

There are two basic methods of creating an image on a CRT. The first is raster scan, the same as used in television and computer screens. The second is stroke writing, which generates a display as you might draw with a pencil. To understand their advantages and disadvantages *consider* how the two different, yet common, displays are generated.

Note:

Raster scanning will be used for Horizon- and *weather radar* background information.

Stroke writing is used for all other parameter drawings on the screen. The permanent change between raster and stroke appears as one picture only.

In all modern display units a combination of the two display techniques is used.

Raster Scanning

As Figure 6 shows, a CRT is an evacuated glass tube, that is designed in such a way that electrons are caused to move along the tube and deflected so as to write across a fluorescent screen, similar to a pencil drawing lines, one below the other down a piece of paper.

Once the last line is drawn at the bottom of the screen, the electron beam starts again at the top. The markings (lines) it has made on the screen will last for a short while as the inner part of the screen is fluorescent.

The beam writes lines very quickly, too quick for the human eye to see, and by adjusting the *density* of the electron flow as the beam moves across the screen different *density* lines can be imaged and a picture can be produced. A moving picture is possible as "re-write" times are fast.

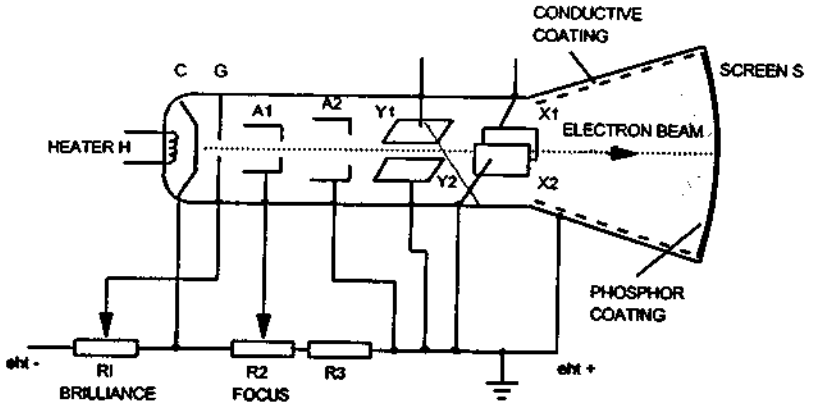
Figure 7 shows a working cross section of a CRT with electrostatic *deflection* and a voltage divider which *provides appropriate* validates from a voltage supply.

The heater (H), is an electrically heated tungsten wire inside the cathode (C). It is insulated from the hollow cylindrical nickel cathode, and when a current is passed through the heater, it raises the temperature of the cathode to a point where the electrons become agitated and the cathode *emits* electrons (thermionic emission).

The negatively charged electrons are attracted to and accelerated towards the anodes A1 and A2 (these are usually cylinders with holes at either end and are positive with respect to the cathode, A2 more so than A1). The electrons are attracted to the anodes, and because of their acceleration the electrons move through the centre of the anodes to impinge on the screen.

The grid G, also a hollow cylinder, has a negative voltage which can be varied by R1. *Varying* R1 can make the voltage on the grid more or less negative to the cathode thus controlling the amount of electrons and thus the brilliance of the display.

Figure 7: Working Section of a CRT



Focusing is *achieved* by altering R2, thus altering the voltage between A1 and A2. When the electrons strike the screen S, which is *coated* with a phosphor coating, it causes the phosphor to luminesce and give a spot of light on the screen.

There is a return path for the electrons from the screen to the cathode, otherwise unwanted negative charge would build up on the screen. This does not happen because when struck by electrons, the screen *emits* secondary electrons, these are attracted to and collected by a *conductive* coating (graphite) on the inside of the tube and returned to the cathode via the power supply.

To *trace* out a display it is necessary for the spot to be deflected horizontally and vertically. Figure 7 shows an electrostatic *deflection* system where the two sets of plates y1 and y2 and X1 and X2 deflect the beam horizontally and vertically. The y plates deflect the beam vertically and the X plates deflect the beam horizontally, by making one plate of the pair more positive than the other.

By combining the two effects the beam can be made to move to any position by controlling the polarities of the two sets of plates.

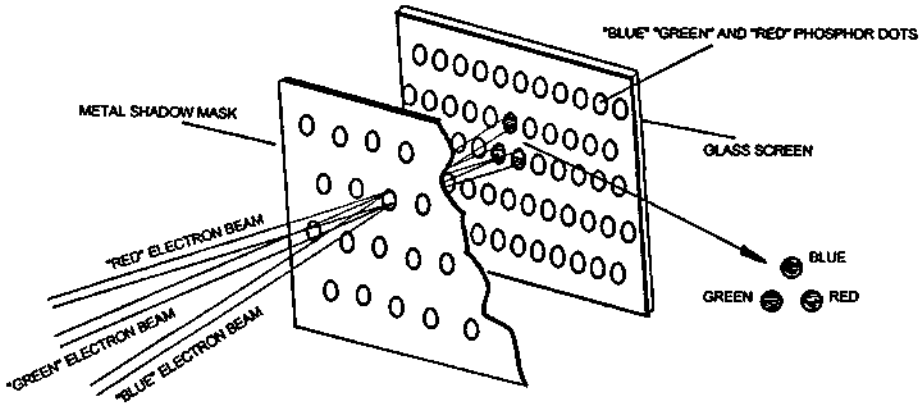
To produce a picture on the screen it is scanned from left to right and from top to bottom. Similar to the way you read a book. The left to right sweep is controlled by a timebase which *consists* of an amplifier and relaxation oscillator. The timebase applies a *varying* signal to the X plates (electrostatic *deflection* system) or coils at the top and bottom (electro-magnetic *deflection* system).

Principle of the Colour CRT (stroke writing)

Colour CRT's are *widely* used in aircraft cockpit displays - particularly for the larger aircraft though some cargo type aircraft use a green screen flight deck.

Unlike the black and white CRT the colour CRT has three electron guns.

Figure 8: Principle of the Colour CRT



The CRT has 3 electron guns each dedicated to a colour: red, green or blue.

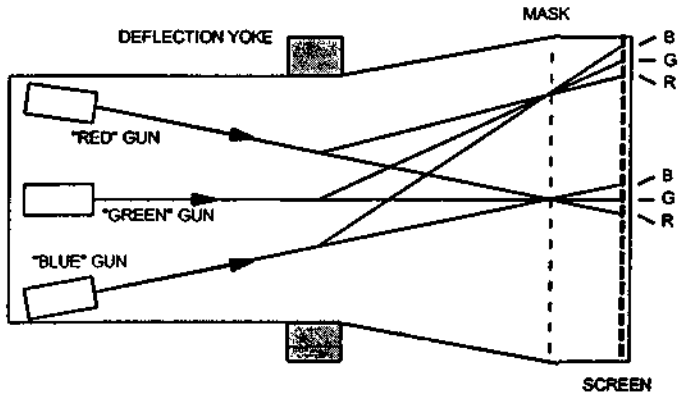
The inside of the screen is *coated* with many thousands of tiny *dots* of red, green and blue phosphors, arranged in small areas, each area *contains* a phosphor of each type. The beam from a particular gun must only be able to strike the designated phosphor, i.e. the 'red' gun electrons strike only the red *dots*, the 'green' gun the green *dots* and the 'blue' gun the blue *dots*. To achieve this a perforated steel sheet called a shadow mask is *accurately* positioned adjacent to the screen.

This shadow mask is very *accurately* manufactured with small holes through which the beam paths pass through.

As the three electron guns scan the screen under the influence of the *deflection* coils, the shadow mask ensures that each beam strikes only its designated phosphor. By independent control of the three guns and their beam currents other colours may be generated. If only the red, green, or blue colours is required then, as the scanning is so fast, you would see a completely red, blue or green screen.

If all three guns are operating with an equal mix of red, green and blue this would give a white *trace*, if only red and blue were *emitting* the mix would be a *purpur trace*. If the green gun electron beam and red gun electron beam current was higher than the blue then a yellowish *trace* would appear. So by controlling the intensity of the three electron beams various colours may be obtained.

Figure 9: Operation of the Shadow Mask



Advantages of the CRT

Most CRT technology was developed for television receivers. For the first EFIS (electronic flight instrument system), smaller CRT's of high brightness were developed. The tubes have *deflection yokes*, where colour *purity adjustments* are made at the factory and bonded in place. With the growing popularity of EFIS displays, newer CRT displays are as large as a small television receiver.

Advantages of the CRT:

- Full colour graphics display
- Good *resolution*
- Sunlight readable
- Dimmable
- Reasonably power efficient

- Relatively inexpensive
- *Wide* temperature range
- Proven technology
- Life span as good as most technologies

Disadvantages of the CRT:

- Requires several power supply voltage
- Requires very high voltage
- Generates magnetic fields which can radiate
- Is constructed with a fragile glass envelope
- Is heavier than other display technologies
- Requires significant depth behind the front panel

Item 5, on *weight*, includes magnetic *deflection* components and transformers for generating high voltage.

Item 6 on panel depth is the most significant CRT problem. Depth represents wasted volume. Space behind the instrument panel is always tight in an aircraft and small assemblies are highly desirable.

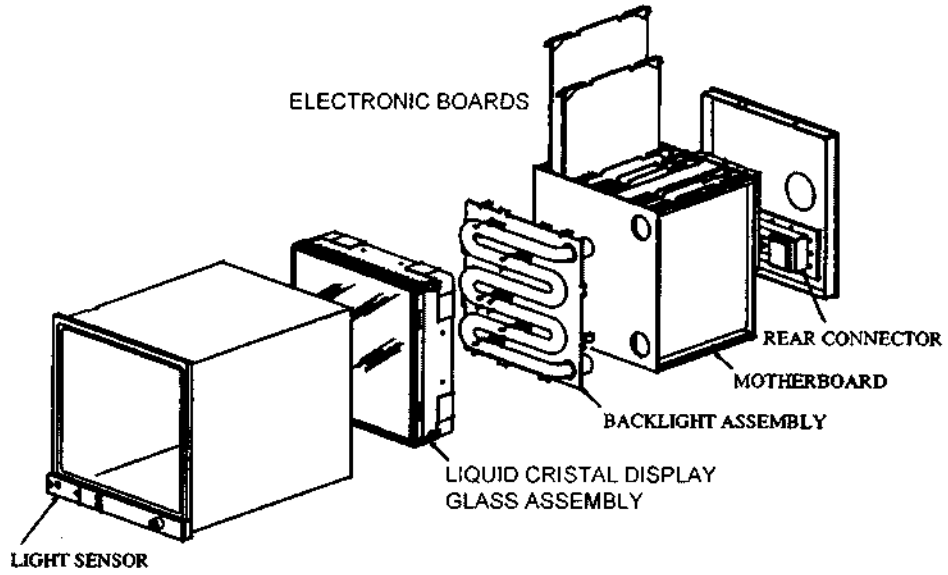
Most EFIS displays in aircraft currently used are CRT - based. The flat panel Active Matrix *LCD* however, is replacing the tubes in many existing and new-design aircraft.

Liquid Crystal Display (LCD)

The newest technology goes toward *improvement* of the display units in the cockpit. This *LCD* units are smaller, lighter and uses less power. There is no radiation of electric or magnetic fields who must be shielded. The display has a better *readability* in bright sunlight and a very high *resolution*, so more information is displayable. As a safety aspect, there is no CRT to *implode* and no more very high voltage are used. The *reliability* is high, there are no *adjustments* for *purity*, *convergence*, focus and *deflection* necessary. The costs of a *LCD* is higher than of a CRT, because it is more difficult to manufacture it. The chance to have failed picture elements (*pixels*) is high, with so many active transistor-elements.

The DU *contains* a display glass *assembly* with *liquid crystal display LCD* elements. About 4 million *thin film transistors* (*TFT*) *absorbing* or *conducting* the light from the bright backlight *assembly*. through the glass panel. Each colour *dot* has 3 transistor elements. One for red, green and blue to produce the desired colour.

Figure 10: Example of an Active Matrix *LCD* Display Unit



Liquid Crystal Technology:

LCD is *unique* in that it can be reflective, emissive or both. The basic function is that of a light valve. The *LCD* turns ambient light on and off or it can be supplied with a light source. Thus, it is reflective or emissive. Figure 11 shows the basic construction, the heart of the *LCD* is a fluid, the *liquid crystal*, between two clear glass plates. The rear glass is *coated* with a thin, transparent metallic film. The front plate is *coated* in specific areas with metallic material. *LCD* construction to this point is similar to the plasma display described.

Outside of the front and rear glass plates is a polarizing film. This creates plane polarized light out of *randomly* polarized light. Polarization occurs when the electric field of the light is parallel to a plane. A polarizing film is like a microscopic window *blind*. When light strikes the polarizing film where the electric field can pass through the *blind*, light passes. Light of other polarization is blocked.

If two polarizing films are aligned so polarization is the same, we see through the two films. On the other hand, aligning the films so polarization is different blocks the light; the result is darkness.

Crystals, *liquid* or solid, have a *unique* characteristic of being able to rotate the polarization of light. Crystals have an ordered *array* of molecules which can have characteristics similar to the window *blind* example above. Another important characteristic is that the crystalline structure reacts to electric fields. In a *liquid crystal*, the ability to rotate the polarization of light is a function of the applied electric field.

Assume the polarizing films are arranged so polarization is the same front and back. Pass light through the films and there would be little *attenuation*. By *applying* an electric field across the *liquid crystal* material in the space between films, polarity of the light is rotated. If that rotation is 90 degrees the area where the electric field is applied becomes dark. Polarized light was rotated 90 degrees so it cannot pass through the rear polarizing film.

Selective application of an electric field, therefore, causes areas to become *opaque*. A light source behind the rear polarizing film generates characters such as 7-segment digits and alphanumeric.

If the polarizing filters are oriented 90 degrees to each other, the display appears dark unless there is rotation in the space between front and back polarizers. In this case, the electric field causes a bright area, as the additional 90 degree rotation allows light to pass through the *LCD*. By orientation of front and back polarizing films, the *LCD* can be configured as a white on black display or black on white.

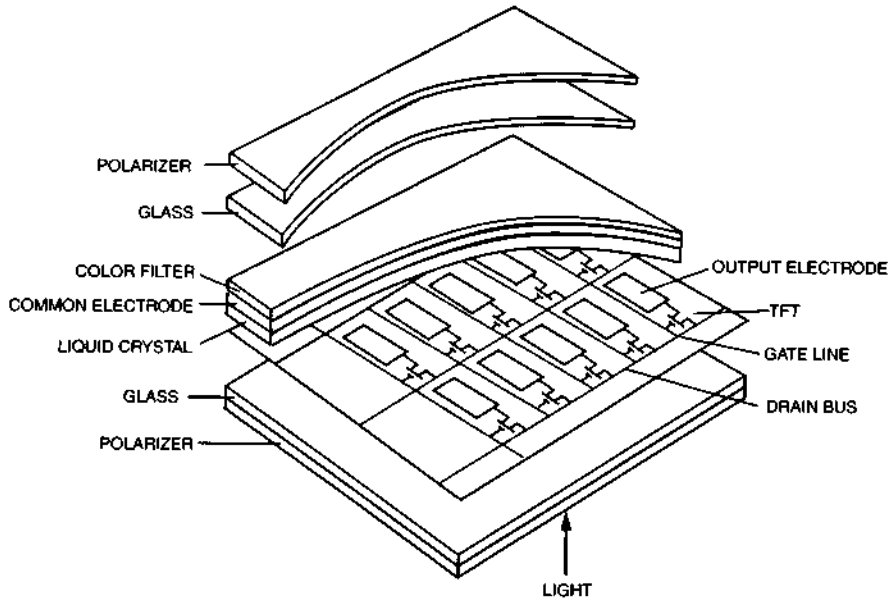
A static electric field produces the desired results but continuous application of an electric field can cause unwanted chemical reaction in the *liquid crystal* material. To prevent this, the applied field alternates, with no DC component.

The *LCD* is a shutter, allowing light to pass or be blocked. If a light source is placed at the rear it *emits* light and becomes an emissive display. If the glass panel is replaced with a mirrored surface, ambient light is reflected from the rear glass plate and no illumination is required. Here, the display is a reflective type. If the mirror is partially transparent, the light can be reflected or transmitted.
This is called transflective.

A major advantage of *LCD* is low power. Only an electric field is required to pass or shut off light. Virtually zero energy achieves this change of *state*. When an *LCD* is in the reflective mode, a low power display *provides* sunlight readable results with nearly zero power. With back lighting, energy is required.

For a sunlight readable emissive display, the light energy is significant and only highly efficient back lighting may be used, such as fluorescent. *Incandescent* lighting may be used for night-time viewing. A common arrangement is a transflective display with *incandescent* lighting for night, reflective mode for sunlight and emissive mode for night viewing.

Figure 11: Principle of an Active Matrix *LCD*



Because power to the *liquid crystal* is very small, a large number of segments may be used. As pointed out for the LED, a large *array* has problems with the amount of energy in a small area. A bright LED consumes 10 to 20 mW and, for a large matrix such as 1024 elements, all segments on can dissipate 20 watts.

Clearly, 20 watts cannot be highly concentrated in the front panel of an avionics display. On the other hand, 1024 *LCD* segments may occupy a small area with no danger because power is extremely low.

What does become a problem, however, is how to connect 1024 segments to the driving electronics. Graphics displays produce pictures with a large number of *emitters* or light shutters. As an example, a reasonable picture can be made with a matrix of 256 by 256 picture elements, or *pixels*. This represents 65,536 *pixels*. The word "reasonable" is used in this example; a picture of 256 by 256 *pixels* is not television quality, which is approximately 500 by 750 *pixels*. A good quality computer monitor is 1024 by 1024 or more than a million *pixels*. It is not possible to have a million-pin connector to connect the *LCD* to one million drivers on a print-

ed circuit board. Before generating *LCD* graphics, connections need to be reduced to individual *pixels*.

The technique is to place the driving electronics on the glass with the segments. Very small *thin film transistors*, *TFT*, are deposited on the glass and connected to the segments. Each *pixel* is addressed sequentially. For a large *LCD* with a 1024 by 1024 matrix, each *pixel* is addressed by horizontal and vertical lines where the lines intersect. When segments are addressed, a transistor acts as a storage element to keep the segment activated until addressing returns to that *pixel*. With X-Y multiplexing, the number of lines addressed is now 2048, still a large number. A 2048 pin connector is impractical and further reduction of interconnects is required.

By mounting integrated circuits directly to the glass and using electronic multiplexing, the number of interconnects is reduced to a handful. This display is called the active matrix *liquid crystal display*, or AM*LCD*.

The shape of a *pixel* is *considered* to be square. If a display has an aspect ratio of 4 to 3, meaning *width* is 4/3 that of height, *pixels* are arranged as a matrix of 4N columns by 3N rows. If a display has a *resolution* of 1024 *pixels* along the horizontal, it requires 4N columns to be 1024 or N equals 256. The number of rows is 768. The total *pixels* is 1024 X 768 or 786 X 432 *pixels*. Another way of describing this display is to specify a *resolution* of 1024 X 768, which may be *recognized* by as a computer monitor specification.

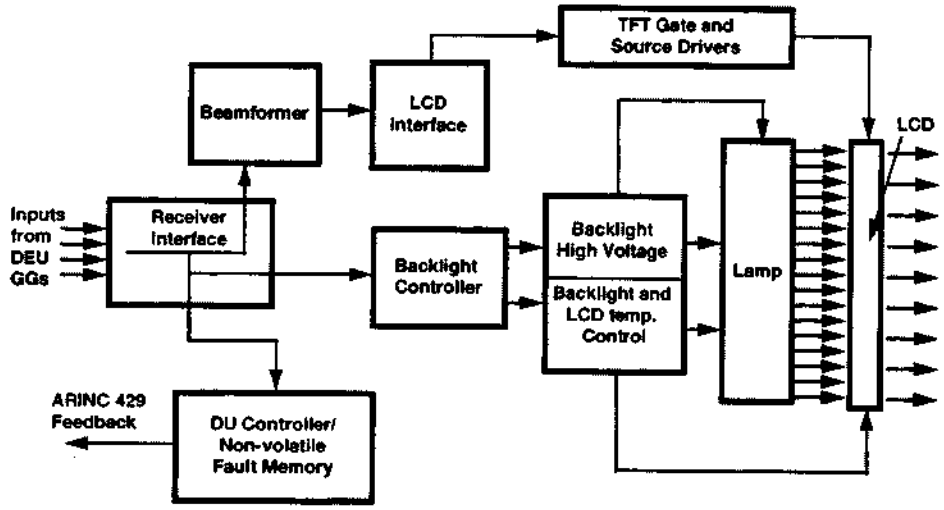
LCD's generate colour graphics by providing three light shutters, one for each additive primary colour at each *pixel*. Every shutter also has a microscopic filter; red, green or blue. This implies that if there are 1 million *pixels* in a 1024 by 1024 display, there are 3 million *LCD* light shutters!

What is the probability of getting 3 million *pixels* to work? It is small, and there is a lot of scrap in the AM*LCD* industry. The product, therefore, is expensive. This is changing as manufacturing techniques grow more refined *LCD* displays have limitations. Most early problems have been solved but one remains: Temperature range, an *LCD* can be destroyed by *excessively* high or low temperature. Low temperature freezes the *LCD* material and causes possible permanent damage. Response time also slows. For an instrument display, where there is no high-speed motion, slowness can be tolerated. At a low temperature that renders an *LCD* television worthless, the *LCD* display is satisfactory for an indicator. Noticeable slowing occurs at temperatures below about 10 degrees C and freezing occurs below -20 degrees C. Permanent damage results below about -40 degrees C. With the possible exception of bush planes in Alaska, aircraft are not usually flown with a cockpit temperature below -20 degrees C. And aircraft are rarely stored below -40 degrees C.

Back-lighting *LCD* displays is an art. To be sunlight readable requires significant intensity. Any backlight must be reasonably efficient if the display is not to get too warm. For colour *LCD*'s, the backlight should have a white colour that does not shift with dimming, as would an *incandescent*. Of all the potential lamp technologies, fluorescent tubes emerged as the choice of illumination. Specialized fluorescent tubes with steady illumination are mounted behind the *LCD*.

Fluorescent tubes require high voltage and a "kick" to start an internal arc. The lamps are a constant current device so they need current limiting. A power supply is usually included as a part of the lamp *assembly*. In spite of the fact that fluorescent tubes are more efficient than other lamp technologies, they dissipate energy. The power supply also needs to dissipate heat so backlight assemblies often have heat *sinks*.

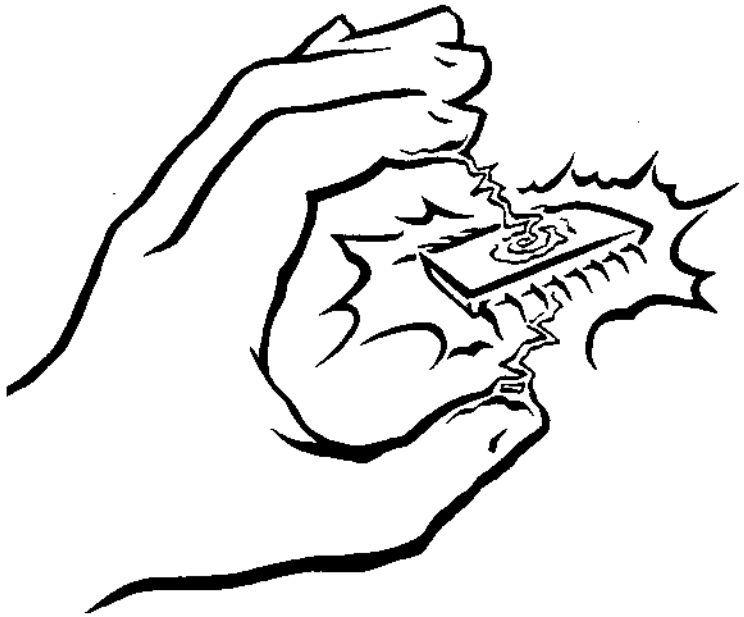
Figure 12: Example of a Cockpit LCD Display Unit Block Diagram



5.12 Electrostatic Sensitive Devices

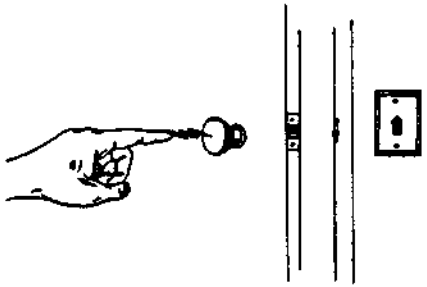
Electrostatic Sensitive Devices (ESD)

Figure 1: Explosion caused by Electrostatics on an Integrated Circuit



Electrostatic discharge, also known as ESD, is defined as the "equalization of static charge between two surfaces." Most of us can relate to ESD in a more personal way. It's the small shock you get when walking across a carpet and then touching a metal object, like a TV or doorknob.

Figure 2: Typical ESD Event



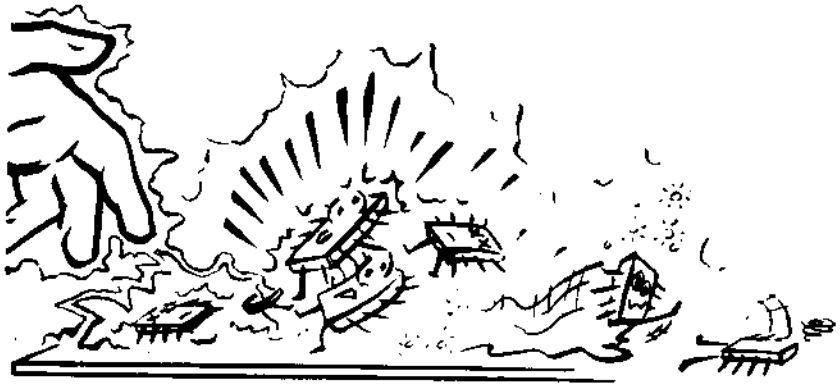
That small shock is an ESD event!

This small shock is exactly the same as the large bolts of lightning you may see during a thunderstorm. The ESD event that you may feel when walking across a carpet and touching a metallic object measures between 12,000 and 40,000 volts. The common *level* a human can feel is from 3,000 to 4,000 volts.

ESD can damage most electronics

Static can damage, degrade or "blow" electrical components. To increase the *reliability* of the electronic systems, we must be very careful to handle with it. Electrical components *contain* very small *conductors* that are sensitive to ESD

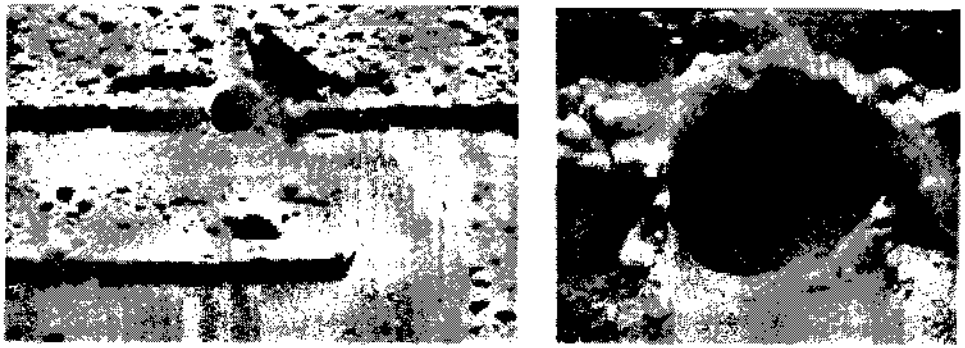
Figure 3: Damage on Integrated Circuits.



Damages caused by ESD

Some *conductors* are only one millionth of one millimeter in diameter (smaller than a speck of dust). These *conductors* melt and/or vaporize when an ESD event occurs

Figure 4: ESD - Damage inside integrated circuit (Magnification 1000x and 5000x).



Sometimes an ESD event causes a hidden or latent failure. This type of ESD damage will not appear in the electronics during testing of final check-out, but because of stresses of in-operation electrical overstress, power on/off, etc...this component fails at a later date.

Figure 5: Consequences of ESD.

Latent failures affects the performance of electronic systems and in our case the *reliability* of the avionics-systems in an ESD event airplane.

ESD sensitivity *withstanding* voltage (Human Body Model HBM)

Type of *semiconductor*	Voltage
Very sensitive parts	10 – 100
Power MOSFETS	100 – 300
VLSI before 1990	400 – 1000
Modern VLSI	1000 – 3000
CMOS	1000 – 5000
Linear MOS	800 – 4000
Transistor old	600 – 6000
Transistor modern	2000 – 8000
Power Transistor	7000 – 25000
Film resistor	1000 – 5000

ESDS Part Label

Electro Static Discharge Sensitive (ESDS) parts and package material are usually labelled with stickers of the colours black and yellow with a symbol and text as shown in Figure 6.

Figure 6: ESDS Symbol

Generating static charges and damaging fields

Everyone who handles with electronic system is responsible to ensure that sensitive components are protected from ESD damage. For this you must become familiar with Static Generators. There are many ways static charges damage ESD sensitive components. The most common method of generating static is by rubbing or sliding two different materials together. This is called Triboelectric charging. Tribo means "to rub." The size of the charge generated depends on the type of materials rubbed together. High static-producing materials include plastics and glass. Even air may become charged when it moves through an area. One of the worst static generators is tape. Unrolling a 10 cm piece of tape can generate between 3,000 and 5,000 volts of static.

Figure 7: Generating Electrostatics

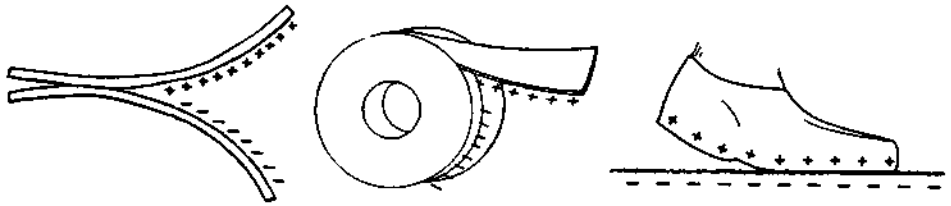
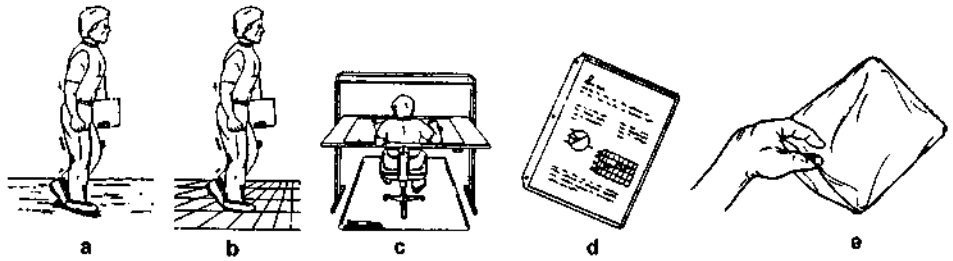


Figure 8: Typical Electrostatic Voltage during your Work



a) Walking over carpet 1'500 - 35'000 Volt

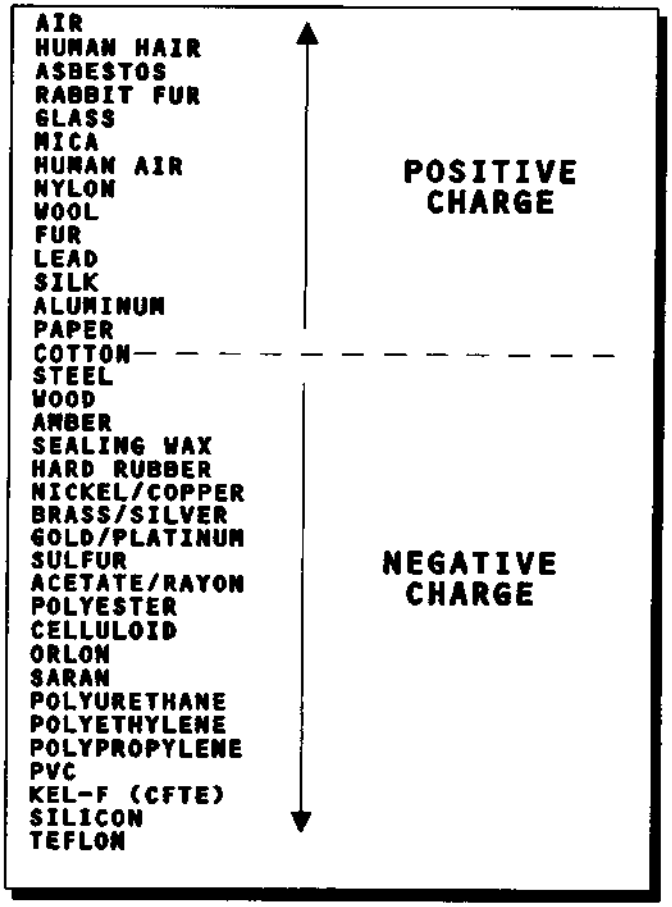
b) Walking over Vinyl floor 250 - 12'000 Volt

c) Working at unprotected bench 700 - 6'000 Volt

d) Putting work-instruction in a vinyl-cover 600 - 7'000 Volt

e) Plastic bag picked up from workbench 1'200 - 20'000 Volt

Triboelectric series



ESD Training

ESD damage can be prevented. One way is to *provide* a discharge path to ground the charge. How to get grounded!

Wear a snug-fitting wrist strap that is connected to ground. Your wrist strap should always touch the skin. It should not dangle freely. Don't wear a wrist strap inside out, over clothing or just held in the hand. When worn this way, the strap cannot prevent ESD damage.

Figure 9: Wrist Strap

Wrist straps are required for all visitors who are at a work station and handling ESDS material/assemblies. Check your wrist strap daily. Personnel shall not be connected directly to hard ground. Wrist straps have a current limiting resistor (one mega-ohm) to ensure that personnel safety *requirements* are met.

Wear ESD shoes or straps when it is *inconvenient* or impossible to wear a wrist strap. Foot straps must be worn on both feet. Foot straps are only effective on *conductive* flooring. ESD shoes or straps must be checked daily

Figure 10: ESD Shoes.

Clothes must be put on and taken off away from the static-safe work station.

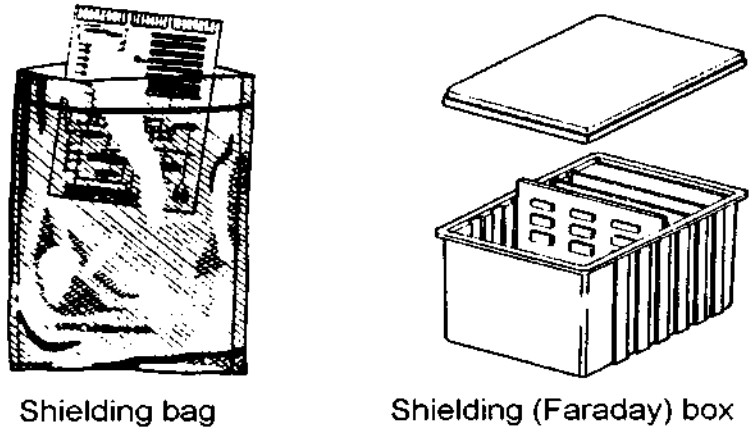
Personal grooming should not take place within one meter of an ESD work station or ESDS material.

Material handling

ESD-sensitive material should be handled by trained personnel at a certified work station. All ESDS material should be packaged and transported in a Faraday Cage static shielding enclosure. Examples of Faraday Cages are static shielding bags that have been sealed with a ESD caution label on the bag or a covered *conductive* tote. Keep units not being worked on in their protective packaging. Remove ESD sensitive material from its protective packaging only when it is absolutely required.

All ESDS material must be re packaged in a Faraday cage before going to breaks or lunch.

Figure 11: ESD Package Material



Shielding bag

Shielding (Faraday) box

Figure 12: Warning Plates

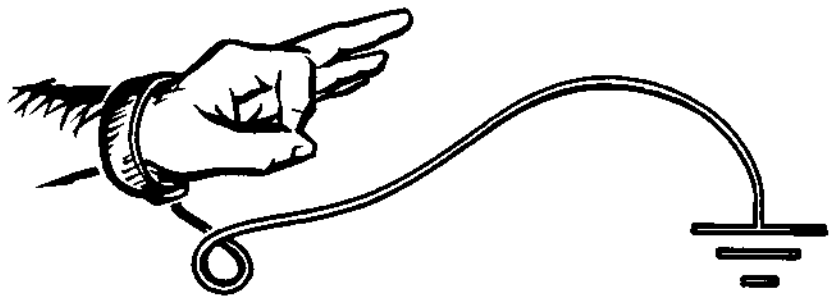
Work stations

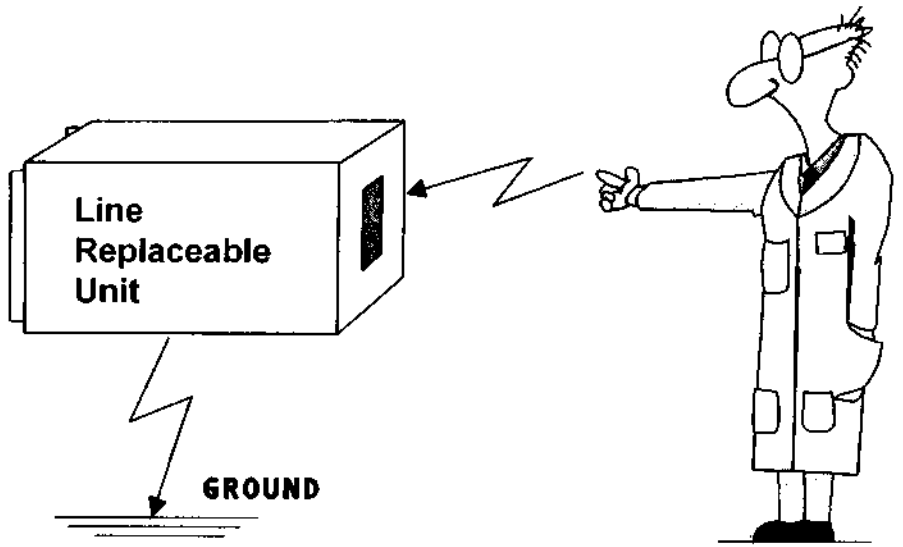
Handle ESD-sensitive materials at certified work stations only. When you are seated at the work station, your wrist strap should be attached to the grounding jack at all times. Static generators like paper and plastic should be kept a minimum of 30 cm from the ESD sensitive assemblies. ESD-safe work stations are labelled. Check and make sure that the *conductive* mat is properly grounded. Use a topical approved spray cleaner to keep your mat clean. Then use a topical antistatic solution to preserve the dissapative qualities of the mat. Ionizing fans are used to help eliminate the build-up of charges on plastics that cannot be eliminated from the work station. Only tools required to perform the operation will be on the work surface. Tools having plastic (insulated) handles must be treated with a topical antistatic. Whenever possible, all static generators such as untreated plastic, paper and rubber should be removed from the work station. All electrical equipment used at the work station shall be properly grounded. Electrical equipment must have a 3-prong plug.Personal items are not *permitted* on ESD work stations. Sweaters, coats, smocks, etc., are not *permitted* on the back of the chair.

Figure 13: ESD Safe Workstation

Things to remember to prevent an ESD event:

Figure 14: Grounding





At aircraft

- Establish good grounds (as for fueling) if a unit has to be serviced.
- Discharge your own body electricity by touching the metallic aircraft structure or by using grounded wrist band.
- When connecting test equipment, first establish the ground connection.
- Avoid using flying leads on pins for testing.
- Use approved test equipment only.
- Do not touch any connectors pins
- Protect the unit by placing protective caps on the connectors.
- Place the unserviceable unit in its reusable shipping container.

At workstation

- The Line Replaceable Unit (LRU) can only be opened at an ESD-controlled workstation.
- The technician has to wear a grounded wristband.
- All test equipment and tools are grounded.
- Removed circuits board are stored and shipped in *conductive* plastic bags or wrap.

Figure 15: Warning Plate on ESD Units

5.13 Software Management Control

Introduction

Each digital LRU (Line Replaceable Unit) *consists* of Hardware, the electronic devices, and Software, instructions that tell a computer what to do. **Software** *comprises* the entire set of programs, procedures, and routines *associated* with the operation of a computer system.

With the *considerable* use of software on modern aircraft used in essential systems such as flight controls, engine controls, electrical generation, navigation flight instruments and auto-flight it is essential that the software design must be investigated and control of its certification maintained.

The basis for the certification of software in aircraft equipment and systems is RTCA (Radio Technical Comission for Aeronautics) document DO-178B, and EUROCAE (European Organization For Civil Aviation Equipment) document ED-12/12B entitled "Software *Considerations* in Airborne Systems and Equipment Certification".

These documents *provide* guidelines for the production of airborne systems equipment software. They are used internationally to specify the safety and airworthiness of software for avionics systems. They describes techniques and methods *appropriate* to ensure the integrity and *reliability* of such software.

Software Definition

LRU's of new generation often *contains* their software in different packages, which can be divided into different categories:

- Core Software or System Software
- Operational Software or Application Software and
- Data Base Software.

The Core Software refers to the operating system and all utility programs that manage computer resources at a low *level*. It also defines the interface of this specific LRU to other LRU's.

The Operational Software, or Application Software, defines the part of a computer program, that varies for different aircraft configuration or changes for each different airline.

By *implementing* a new Data Base Software, only parameter values will be changed. Therefore it is not a program change. An example would be the navigation data base, that *contains* flight plan information such as runway direction or landing system frequency etc.

In the RTCA Document DO 178B, software *levels* are defined from A down to E:

- *Level* A, where anomalous behaviour can cause a catastrophic result down to
- *Level* E, where anomalous behaviour will not effect the safety of the aircraft.

Software Handling

Depending on the Software *Level*, different care must be taken in documentation and handling of the software. Only authorized personal may modify the software which is classified in *level* A. Software which is classified in *level* E can be done by maintenance personal, but only if respective documentation is available.

In any case it must be guaranteed that:

- only the authorized software will be influenced by the load-activity,
- the successful loading must be acknowledged,
- no other systems will be affected.

The Core-Software should normally never be touched, because it is a part of the control loop of the aircraft. An uncontrolled change could have a catastrophic effect on the aircraft. A change of this software can only be made with the agreement of the aircraft manufacture and the LRU manufacture. This is documented in authorized Service Bulletin (Cover-S/B) of this two manufactures.

A change of the Operational/Application-Software needs also the agreement and documentation of both manufactures. But the LRU-Manufacture can be bypassed, if the airline engineering guarantees an '*equivalent-level*-of-safety', a very complicate act.

A Data Base Software change can normally be done without activity of the manufactures and if it is guaranteed that the software is classified in *level* E.

5.14 Electromagnetic Environment

Introduction

With the introduction of computers and Fly-by-Wire on such aircraft as the Airbus A320, A330, A340 and the Boeing 777, regulations were introduced to increase the protection *levels* of certain systems from the effects of High Intensity Radiated Fields (HIRF) and lightening strikes. These regulations require that critical and essential systems be protected to higher standards than was the case *previously*.

The regulations also require that a maintenance programme be put into effect to ensure the continued airworthiness of such systems.

The prime *requirement* for an aircraft's computer and data transmission system is *reliability*. The units are put through severe temperature, pressure and humidity tests during design and manufacture to check that they work satisfactory in adverse conditions.

It is equally important to check the electromagnetic *compatibility* (EMC), i.e. that it radiates little or no electromagnetic interference to cause problems to other systems and is also unaffected by external electromagnetic interference from other systems.

High Intensity Radiated Fields - HIRF, Electromagnetic Interference - EMI

Electromagnetic interference (EMI) may come from inside the plane or from outside it. What makes the internal sources a matter for concern is that they are so close to the systems they might affect; what makes the external sources a matter for concern is that, *despite* their distance, their power *level* can be very high. Although the internal and external overlap in their effects, in general the external sources involve much higher power *levels*, even after travelling some distance; *hence* they may have more serious effects called HIRF's -- which sometimes stands for High Intensity Radiated Fields and other times High Intensity Radio Frequency -- the external signals come either from huge ground transmitters such as radio, radar, and television antennas, or airborne transmitters such as high-powered radar and radio on military planes.

It is similar to the electromagnetic fields induced by lightning and can also affect the proper functioning of critical and essential systems. Low-intensity RF can originate from personal electronic devices (PED) such as Notebook computers and cell phones used by passengers in flight. These low-intensity devices can also affect critical and essential systems. Electromagnetic interference from PED's is suspected as the cause of many unexplained flight control upsets.

Electrical signals are *susceptible* to voltage *transients* caused by lightning and high-intensity radiated fields (HIRF). The airplane critical flight control system, as well as all lightning/HIRF critical and essential systems, must be protected from these voltage for the life of the airplane. Boeing and Airbus *provide* the initial protection in the airplane structure; shielding all cabling is additional protection. Operators are responsible for maintaining the protection by adhering to grounding practices for all components and inspecting the integrity of the shielding and shielding connections.

Mitigation of Damage to Airplane Systems

If electronic equipment needs to be operated in a region subject to changing electromagnetic fields, and if the currents generated by these fields are *considered* harmful, the recommended approach to *mitigating* the harmful effects is to shield and ground the electronic equipment and the interconnecting wiring. As a result, electrical currents generated by lightning or HIRF then circulate through the equipment enclosure to ground without affecting internal circuitry. This enclosure practice extends to interconnecting wiring through the use of cable shielding; that is, the shield is the enclosure that is grounded. Other damage *mitigation considerations* include the location of the equipment and wiring, use of effective wiring, use of good grounding practices, and building equipment to *withstand transients*. All these tactics are incorporated into the design of modern airplanes and the installed equipment.

A new shielded cable properly installed will exhibit a certain amount of resistance in the shield circuit. By monitoring this resistance, maintenance personnel can determine the ability of the shield to protect internal wiring. Any increase in resistance indicates that a problem is occurring in the circuit, such as corrosion at a junction or loose hardware. When the resistance reaches a certain *level*, maintenance personnel must take corrective action, usually by cleaning the affected junctions, securing loose connections, or replacing the cable.

Low Intensity Radiated Fields

Electromagnetic Interference from Portable Electronic Devices

Operators of commercial airplanes have reported numerous cases of portable electronic devices affecting airplane system during flight. These devices, including Notebook and palmpilot computers, audio players/recorders, electronic games, cell phones, compact-disc players, electronic toys, and *laser* pointers, have been suspected of causing such anomalous events as autopilot disconnects, erratic flight deck indications, airplanes turning of course, and uncommanded turns.

Aircraft manufacturer recommend that devices suspected of causing these anomalies be turned off during critical stages of flight (takeoff and landing). The company also recommends prohibiting the use of devices that intentionally transmit electromagnetic signals, such as cell phones, during all phases of flight. The U.S. Federal Communications Commission already prohibits the use of cell phones during flight.

Electromagnetic interference (EMI) from passenger-carried portable electronic devices (PED) on commercial airplanes has been reported as being responsible for anomalous events during flight. The operation of PED's produces uncontrolled electromagnetic emissions that could interfere with airplane systems. Airplane systems are tested to rigorous electromagnetic standards to establish and *provide* control of the electromagnetic characteristics and *compatibility* of these systems. However, PED's are not subject to these same equipment qualification and certification *processes*. Though many cases of EMI have been reported over the years, with PED's suspected as the cause, it has proven almost impossible to duplicate these events.

Protection from HIRF, EMI and LIRF

The systems that require extra protection are specified and include Fly-By-Wire systems, Auto flight systems, data bases, LRU's (Line Replacement Units) etc.

The *level* of protection from HIRF and other electrical sources (lightning strikes etc.) is increased by the use of Fly-By-Light systems which use light as the data transmission medium in place of digital (or analogue) electrical/ electronic signals.

Whilst external electrical inputs to a data transmission line will have an effect on electrical/ electronic based data transmission systems their effect on light based data systems is *considerably* reduced.

Of course, there can always be mechanical damage - from lightning strikes for example - and the data transmission/reception units can be affected by HIRF as they are electronic.

To *provide* the regulatory *requirements* with regards to HIRF / lightning protection the LRU's in modern digital systems have built in filters and filter pins and over voltage protection systems. The wire design includes the following:

a) Single *layer* of braid internal to fuselage.
b) Two *layers* of braid external to fuselage.
c) Extensive use of twisted pairs and triples.
d) No power or signal grounds external to fuselage.
e) Single point grounding for actuator control electronics and power control units and signals.
f) Nickel coating for fly-by-wire shields
g) Back shell and shield termination for connectors
h) Equipment interface protection

Protection for the systems is also *provided* by aluminated honeycomb wall, floor and ceiling panels around the flight deck and electrical equipment bays

Metal Protected Areas

These may include whole parts of the airframe such as the flight deck and equipment racks. Aluminium-*coated* honeycomb structure may be used to ensure that whole areas are encased in a "metal box".

In this way HIRF should not get through to sensitive equipment. Existing electrically *conducting* window coatings (electric heater film elements) on the flight deck windscreens are used as part of the protection system.

Equipment racks may also be "metal box" shielded and special attention is paid to grounding and earth returns - both for power and signal systems.

Wiring

On the Boeing 777 wiring internal to the fuselage have a single braid *screening* whilst wiring outside the fuselage have 2 *layers* with the outer *layer* terminated at the fuselage bulkhead. The braid is tinned copper.

To help reduce HIRF, twisted pairs and triples are used wherever possible.

Transfer resistance of shielding should be in the region of 2.7 mΩ/m and transfer *inductance* should be 1.0 nH/m. Check the manual on your aircraft for specific values.

Plugs and Sockets

These may vary in design from aircraft to aircraft. The design may also be affected depending on where they are fitted in the aircraft, e.g. within the pressure hull, or outside it.

Shielding is *achieved* using mechanical connection of the cable braid to the metal plug / socket body. In some cases individual wire jumper leads are used and mechanically connected from their respective wires to the metal part of the plug / socket by a metal band.

Maintenance Practices

All wiring, plugs, sockets, and *screening* on the aircraft should be inspected as laid down in the maintenance schedule and AMM. Special checks may be required at intervals such as at 4 and 8 years. The inspections are normally visual but will require bonding testing where necessary and the use of specialist test equipment where specified in the AMM.

For example:

- Every 4 years. Detailed inspection of all critical wiring external to the fuselage.
- Every 8 years. General inspection of all critical and essential wiring inside the fuselage.
- Every 8 years. Detailed inspection of some essential wiring external to the fuselage.

General Visual Inspection

This is a general visual inspection using a light source if necessary, to check for damage, deterioration, contamination, signs of burning, security, open circuits, correct *assembly*, correct torque of plugs/ sockets, locking, chafing, corrosion, broken *strands*, broken *screening*, etc. All wires, shields, ground leads, screen jumpers are to be intact. Panels and access doors may have to be removed/refitted. All this will be specified in the AMM and the maintenance schedule. There is normally no routine *requirement* for dismantling.

Detailed Visual Inspection

This involves an inspection similar to above but it will also include the use of specialist test equipment. All metal airframe (and other) *screening* to be secure, undamaged and complete.

Of course, if any item is found to be defective, the defect should be repaired or the part replaced, and the system tested.

Systems Testing

When testing a HIRF sensitive system other systems may produce radiated signals which can be picked up as *noise*. It is important that the AMM is consulted reference the precautions to take prior to testing any system so that other systems and / or aircraft are not affected.

The actual test will depend on the system to be tested but it will be in accordance with AMM and will normally involve a functional test. It may require that other electric / electronic systems are working at the same time to check for any cross-interference. It may require the use of a special test programme in include simulated interference signals (*noise*).

Remember that testing of some equipment on the ground requires safety precautions to be carried out to prevent injury to personnel. This includes safety distances for radar antennae and radio aerials for example. On the point of test equipment, it must be electromagnetic compatible to the aircraft systems to ensure no problems are caused to the system under test or other systems.

5.15 Typical Electronic / Digital Aircraft Systems

Introduction

In the following section, the general arrangement of typical electronic/digital aircraft systems and *associated* BITE (Built in test equipment) testing will be shown.

In most modern aircraft such as Airbus A320, A330, A340 and Boeing 747-300, Boeing 777 and MD11, Onboard Maintenance Systems are installed. These Systems collect failure reports from each system computer BITE and shows the failures on a display in the cockpit or allows printing of the failures on an onboard printer.

System Tests or Return to Service Tests can be initiated from a maintenance device in the cockpit.

BITE Philosophy

A system is composed of LRU's which can be: computers, sensors, actuators, probes, etc. With the new technology, most of these Line Replaceable Units (LRU's) are controlled by digital computers. For safety reasons, these LRU's are permanently monitored, they can be tested and trouble shooting can be performed. In each system, a part of a computer is dedicated to these functions: it is called Built- In Test Equipment. Sometimes, in multi- computer systems, one computer is used to concentrate the BITE (Built- In Test Equipment) data of the system.

BITE

During normal operation, the system is permanently monitored: internal monitoring, inputs/ outputs monitoring, link monitoring between LRU's within the system.

Fault Detection

If a failure occurs, it can be permanent (consolidated) or intermittent.

Isolation

After failure *detection*, the BITE is able to identify the possible failed LRU's and can give a snapshot of the system environment when the failure occurred.

Memorization

All the information necessary for maintenance and trouble shooting is memorized in a Non *Volatile* Memory.

Test

The test function can be divided into 4 groups.

Power Up Test

The power up test is first a safety test. The purpose of a safety test is to ensure compliance with the safety objectives. It is executed only on ground after long power cuts (more than 200 ms).
Its duration is function of the system which is not operational during the power up test.
If the aircraft is airborne, the power up test is limited to a few items to enable a quick return to operation of the system. The typical tasks of a power up test are: test of *microprocessor*, test of memories, test of ARINC 429 and various I/ O circuits, configuration test.

Cyclic Tests

These tests are carried out permanently. They do not disturb system operation. The typical tasks of a cyclic test (also called IN OPERATION TEST) are: Watchdog test (a watchdog is a device capable of restarting the *microprocessor* if the software fails), RAM test. Permanent monitoring is performed by the operational program (e.g. ARINC 429 messages validity).

System Test

The purpose of this test is to offer to the maintenance staff the possibility to test the system for trouble shooting purposes.
This test can be performed after the replacement of a LRU in order to check the integrity of the system or sub-system. It is similar to the POWER UP TEST but it is more complete. It is performed with all *peripherals* supplied.

Specific Tests

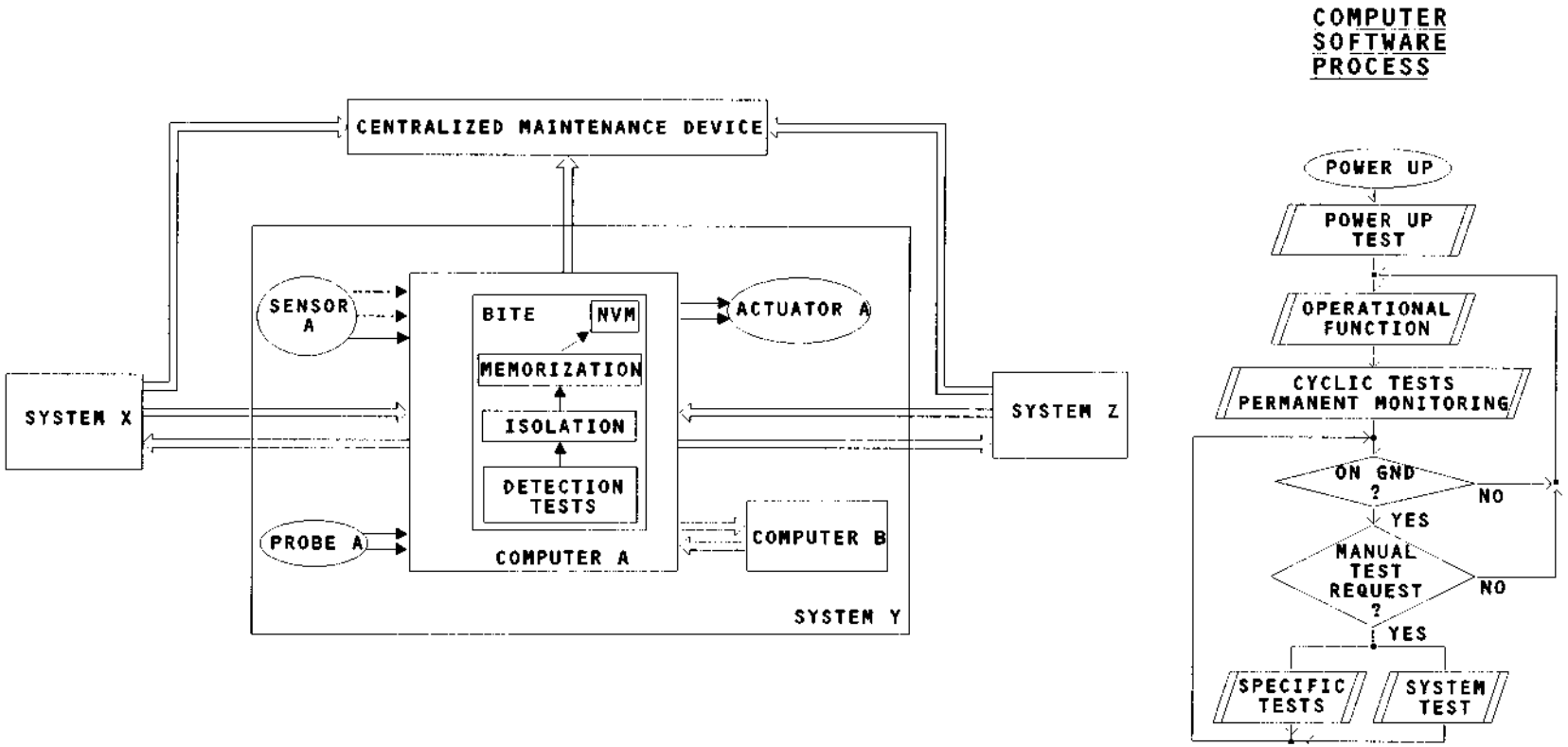
For some systems, specific tests are available. The purpose of these tests is to generate stimuli to various command devices such as actuators or valves.
They can have a major effect on the aircraft (automatic moving of slats or flaps, engine dry cranking).

New Concept

The BITE information stored in the system BITE memories is sent to a centralized maintenance device. The manual tests (SYSTEM TEST and SPECIFIC TESTS) can be initiated via this centralized maintenance device.
Its main advantages are:

- - single interface location (cockpit).
- - easy fault identification.
- - reduction of the trouble shooting duration.
- - simplification of the technical documentation.
- - standardization of the equipment.

Figure 1: BITE Philosophy



ATIMS Air Traffic Information Management System on AIRBUS A330

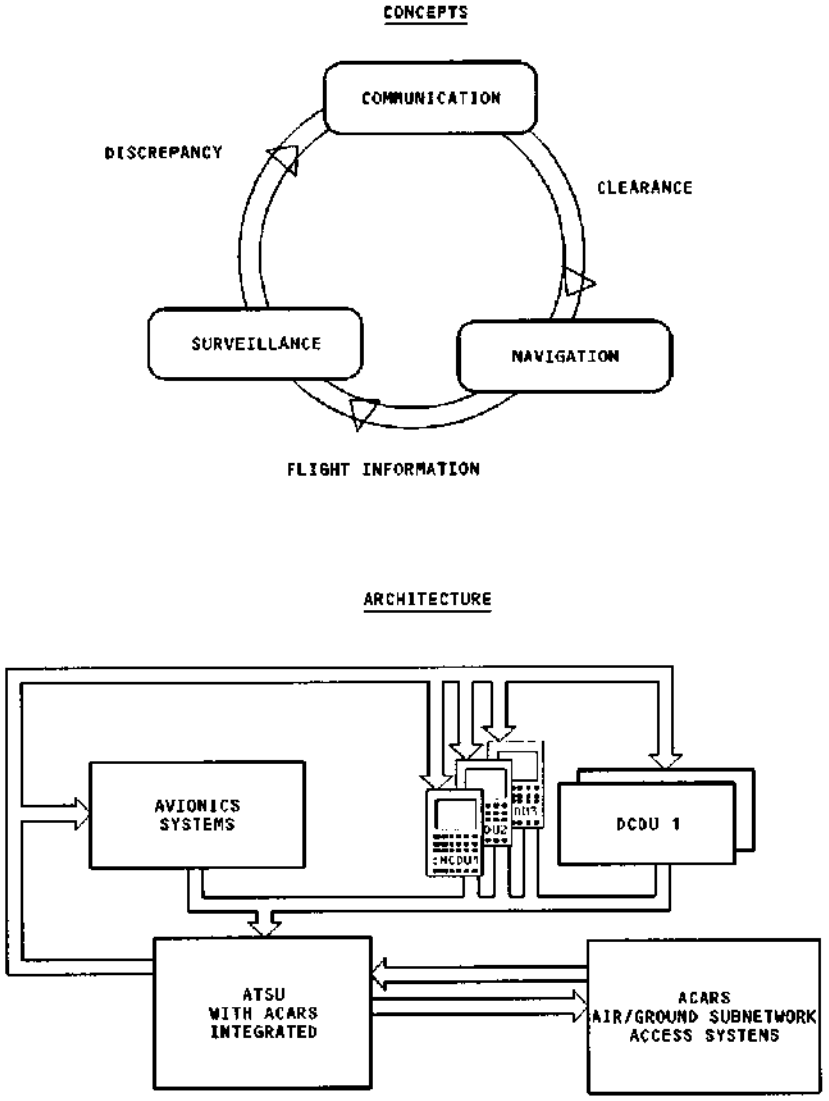
Up to now, flight crews have communicated with air traffic controllers using HF and VHF radio communications which are subject to atmospheric disturbances and so, often difficult to understand.

Furthermore, the transmission networks become *saturated* due to the air traffic increase, and to the limited capability to exchange complex data (routes, weather information...).

Consequently, the Air Traffic and Information Management System (ATIMS) has been developed to enable datalink communications and the exchange of complex data or specific reports between the aircraft and the ground centers:

- controller-pilot datalink communications (HF voice in backup) for air traffic management,
- automatic reporting (position, intention) for air traffic surveillance,
- specific airline-aircraft communications (operational control) to *improve* airline operational costs and flexibility.

Figure 2: ATIMS General Architecture



System Description

The Air Traffic and Information Management System is organized around a host platform which integrates datalink applications and the routing function.

The ATIMS system can be configured in two ways:

- in Pre-FANS configuration:
  - Air/ground communication Router Function (ARF)
  - Airline Operational Control applications (AOC)
    NOTE: The Datalink Control and Display Units (DCDU) and the ATC MSG push button switches are not operational.
- or in FANS A configuration:
  - Air/ground communications Router Functions (ARF)
  - Airline Operational Control applications (AOC)
  - FANS A Air Traffic Control (ATC) applications.
    NOTE: The DCDUs and the ATC MSG push button switches are operational.

System Architecture the ATIMS is composed of:

- an Air Traffic Service Unit (ATSU)
- two Datalink and Control Display Units (DCDU) located on the left and right center instrument panels (not operational in Pre-FANS version),
- two ATC MSG illuminated push button switches (not operational in Pre-FANS version).
- a RESET/ATSU1 circuit breaker to reset the ATSU.

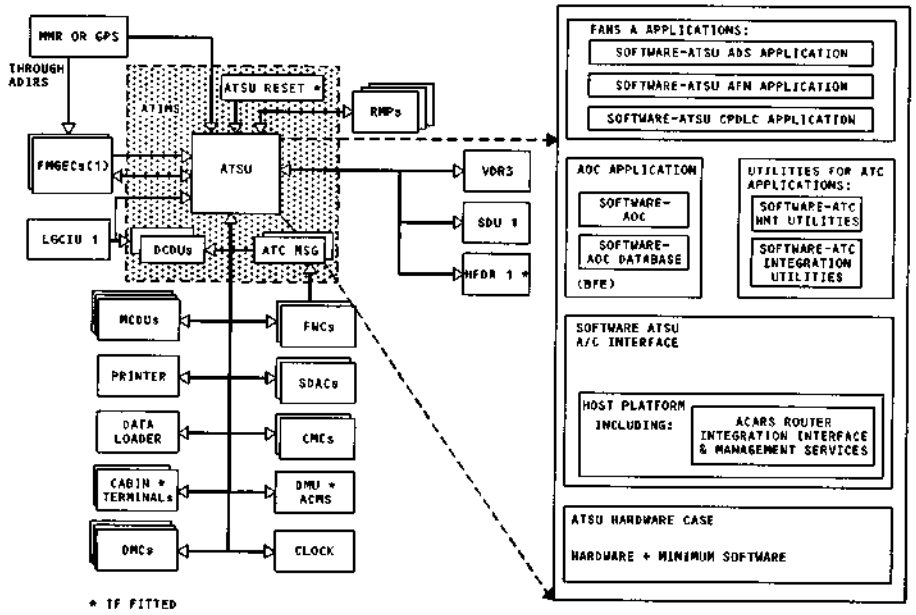
The ATIMS uses the Multipurpose Control and Display Units (MCDU) for maintenance and application purposes.

The ATSU is the main component of the system.

Its architecture is based on:

- an Air Traffic Service Unit (ATSU) hardware case
- an ATSU A/C INTERFACE software uploaded in the ATSU through the Multipurpose Disk Drive Unit (MDDU) or the portable data loader
- an AOC software uploaded in the ATSU through the MDDU or the portable data loader (in Pre-FANS and FANS A configurations).
- FANS A application software uploaded in the ATSU through the MDDU or the portable data loader (only in FANS A configuration).

Figure 3: System Configuration



The main functions performed by the ATSU are:

- to host the various datalink applications, including Airline Operational Control and Air Traffic Services,
- to *provide* management and access to the different datalink services available,
- to *provide* management and access to the various datalink networks available.

1. ATSU A/C interface software
   Its different functions are:
   - monitoring of the system (power supply and BITE functions)
   - *acquisition* of the aircraft parameters for applications software use
   - management of the air/ground communications (ARF function
   - management of the communication with the on-board peripheral units

- management of the human/machine interface (MCDU, DCDU, Printer and alert function)

2. AOC software
The AOC software *consists* in hosted AOC applications which are depending on airline definition. These datalink applications concern operations related to the flight such as flight plans, weather, behaviour of aircraft elements transmitted for maintenance reasons, fuel quantity, personnel management, gate management...

3. FANS A application software The FANS A application packages *consist* of:
   - ATSU AFN (ATS Facilities Notification) application software
   - ATSU CPDLC (Controller-Pilot DataLink Communications) application Software
   - ATSU ADS (Automatic Dependent Surveillance) application Software

   Two utilities packages are necessary for ATC applications:

   a) ATC HMI UTILITIES software: this package *provides* services for the HMI of the ATC application (DCDU, MCDU, printer and ATC alert)
   b) ATC INTEGRATION UTILITIES software: this package *provides* services for integration of the ATC application with on-board system, mainly for interface with the FMS.

   - ATS Facilities Notification (AFN) application
   The purpose of this application is to establish the contact with the ATC ground center, then to *provide* the ATC center with the aircraft registration, the datalink applications available on the aircraft with the *corresponding* addresses.
   - Controller-Pilot Data Link Communications (CPDLC) application The aim of this application is to *provide* dialog between ATC controllers and flight crew, using datalink communication *instead* of voice communications. Each CPDLC message is composed of a set of message elements which *correspond* to the existing phraseology used by current ATC procedures.
   - Automatic Dependent Surveillance (ADS) application The ADS function is to *provide* the ATC ground center with aircraft surveillance data through periodic, event or on-demand reports.

BITE Testing of the ATIMS System

The Basic ATSU interface software enables the BITE Testing of the ATIMS system. Testing is done using the MCDU in the Cockpit. A specific menu on the MCDU gives access to the Test function. Failures of the system components are then collected and reported as Test Result, if no failures are present, a Test OK. message will be displayed.

Electronic Instrument System (EIS)

Another typical digital aircraft system is the electronic instrument system which *comprises* the Electronic Centralized Aircraft Monitoring (ECAM) and the Electronic Flight Instrument System (EFIS). In the following section the general arrangement of the Airbus A320 EIS will be shown.

General

The EIS (Electronic Instrument System) presents on Display Units (DU's):

- EFIS (Electronic Flight Instrument System) information, (i.e. flight parameters and navigation data).
- ECAM (Electronic Centralized Aircraft Monitor) information,

The layout of the 6 DU's and the breakdown of the information displayed on them is presented as follows:

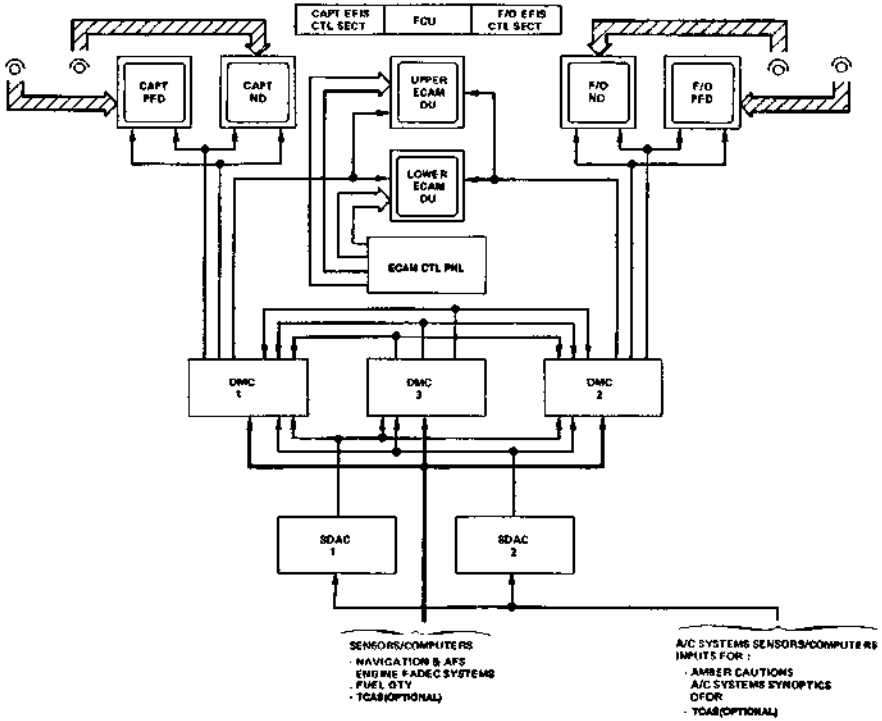
2 DUs are installed side by side in front of each pilot. They display flight and navigation data. On each main instrument panel, in normal configuration, the outer DU will be allocated to the Primary Flight Display (PFD) function, and the inner DU to the Navigation Display (ND) function.

Each pilot is given the possibility to display ECAM information *instead* of navigation information on its inner DU, in order to cover ECAM DU failure cases.

The complete system includes the following components:

- 2 DU's housed in racks on the Captain's main instrument panel
- 2 DU's housed in racks on the First Officer's main instrument panel
- 2 DU's housed in racks on the center instrument panel
- 3 DMC's (Display Management Computers) common to the EFIS and ECAM systems, located in the avionics compartment.
- 2 SDAC's (System Data *Acquisition* Concentrators) located in the avionics compartment.
- 1 dual EFIS/FCU, located on the glare shield, and called the FCU (Flight Control Unit), including CAPT and F/O EFIS control sections.
- 1 ECAM control panel, located on the center pedestal.
- One set of transfer selector switches accessible to the Captain and the First Officer located on the center pedestal. They *provide* EIS reconfiguration controls in case of DMC or ECAM DU failure
- Control/brightness potentiometers for the DUs:
  - those concerning the EFIS DUs are located on the left side of the CAPT and on the right side of the F/O instrument panel (301VU and 500VU)
  - those concerning the ECAM DUs are located on the ECAM control panel
- One PFD/ND transfer push button switch for each pilot, located between the PFD and ND control/brightness potentiometers.
- Two sets of visual attention lights, located on the glare shield, on either side of the FCU, each including a MASTER WARN light and a MASTER CAUT light; these lights are of the push button switch type
- Maintenance and test controls accessible through the MCDU's

Figure 4: gives a complete block diagram of the EIS system.



BITE Testing of the EIS System

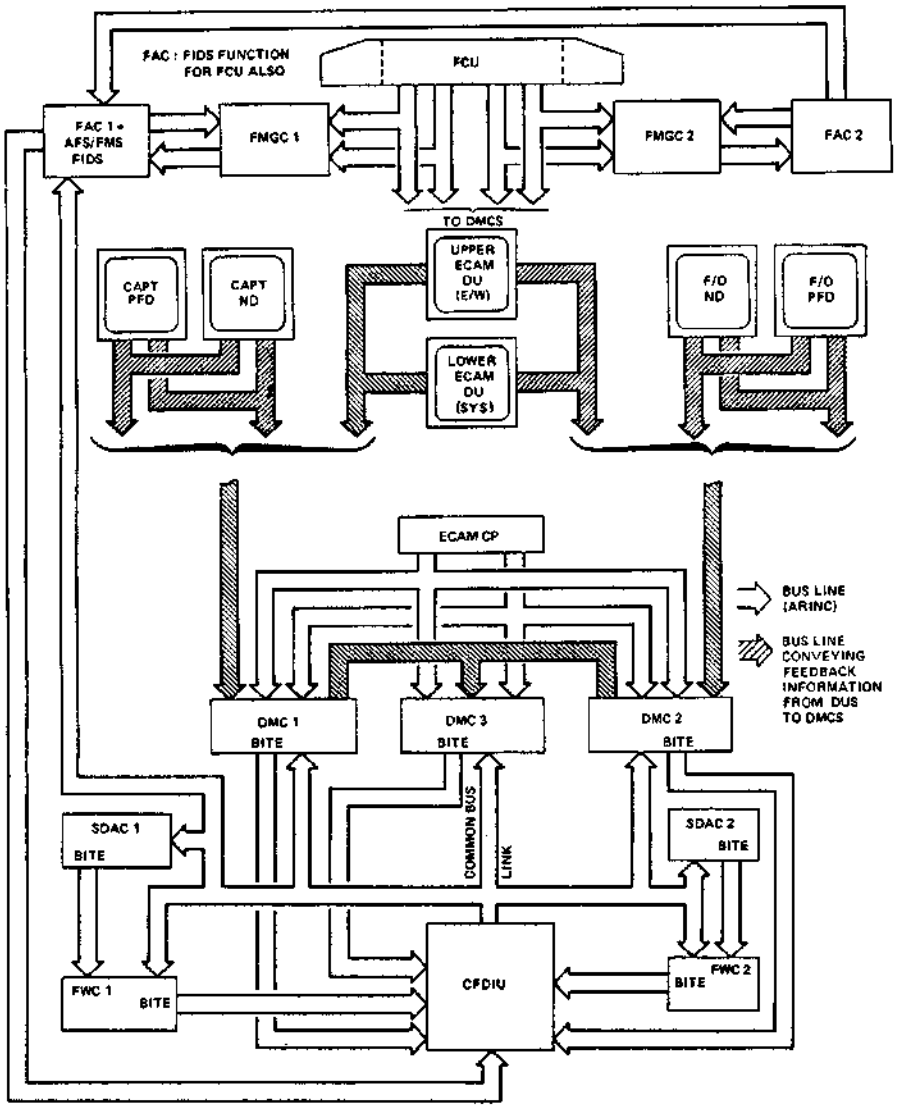
Maintenance and test:

The EIS includes Hard-oriented and Soft-oriented Built-In-Test (BITE) functions. The CFDS (Centralized Fault Data System) facilitates the interface of these BITE resident routines for maintenance and test *implemented* in the various computers.

Failure *Detection* Functions:

The BITE of a computer is able to detect the internal failures as well as failures affecting the I/P parameters. The BITE is designed so as to minimize the undetected failures, and to make the maintenance of the system easier. To that end, the events detected by the BITE (anomaly, abnormal disengagement, failure...) are coded and stored in *non-volatile* memories called BITE memories, under the form of messages in alphanumeric characters. The GMT at the time of each failure is recorded and stored with the *associated* event. The BITE memories store information concerning several flights. The CFDS helps the flight crew and the maintenance personnel by providing the capability of displaying the above messages, as well as system maintenance data and procedures, through the interactive Multipurpose Control and Display Units (MCDU's), located on the center pedestal.

Figure 5: BITE Connections of the EIS System



Flight Management System (FMS) of the Airbus A330

General

The Flight Management System (FMS) performs various functions to help the crew in the management of the flight. These functions are all constructed from a lateral and a vertical flight plan. The pilot can select this flight plan from a data base stored in the system and can modify it at any time.

In the lateral plan, the FMS performs: navigation computation (aircraft position), radio navigation aid selection (automatically or by pilot selection), lateral *guidance* to maintain the aircraft along the flight plan from takeoff to approach.

In the vertical plan, it computes: an optimum speed at each point, other characteristic speeds. Then it computes predictions along the flight plan based on these speeds, weather parameters, and *weight*. It performs vertical *guidance* referenced to these predictions.

Other miscellaneous performance computations are also made.

The crew can insert various data or select function modes through two MCDU's (Multipurpose Control and Display Unit) linked to the FMGEC's (Flight Management, *Guidance* and Envelope Computer).

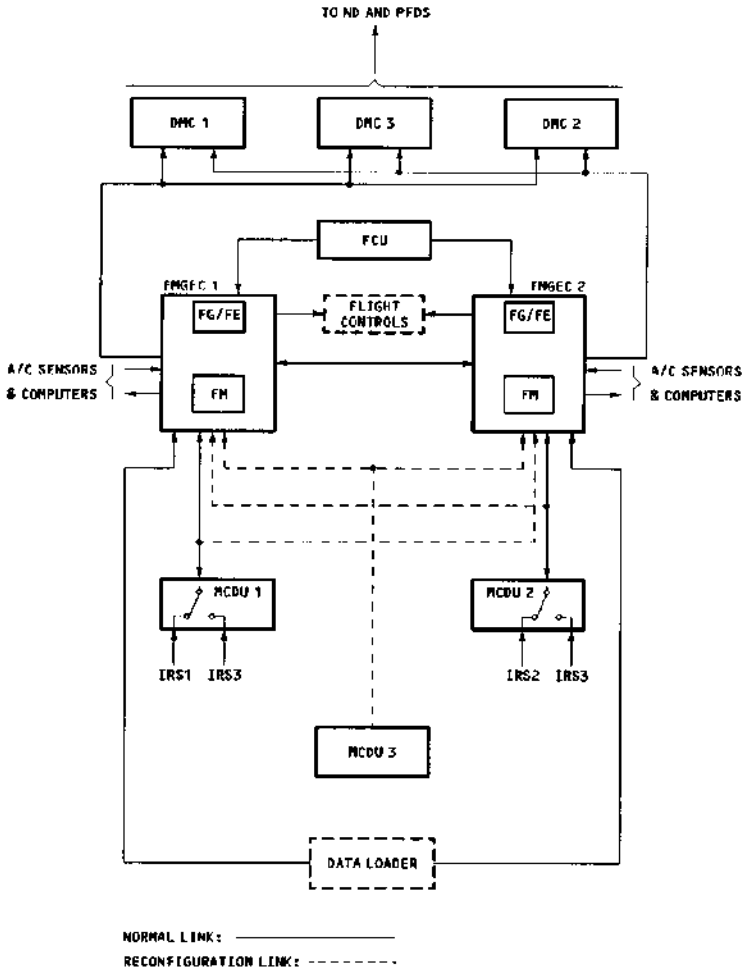
The third MCDU is in hot spare for FM function in case of one of the above MCDU failed. The two MCDUs, both ND's (Navigation Display), and, for some parameters, the PFD's (Primary Flight Display) are also used by the system to display information related to the above-mentioned functions

System Description

The FMS general architecture which shows the two FM portions incorporated in the FMGEC's, with the three MCDUs and the DMC's (Display Management Computer) for display is given in Figure 6. It has to be noted that the navigation data base must be loaded once every 28 days by a loader which is plugged to the FMS in the cockpit.

The MCDU is a colour Multipurpose Control and Display Unit capable of communicating with two FMGEC's and up to other subsystems. The MCDU also *contains* a Stand-by Navigation function which enables it to be used for IRS based navigation. The Stand-by Navigation mode allows limited lateral flight planning within the MCDU that can be used to drive the Navigation Display, *provides* relative lateral path position information and auto-sequencing of the active flight plan. The Stand-by Navigation flight plan reflects, as much as possible, the active FM flight plan upon initial activation. The MCDU design also includes BITE functions to continuously ascertain the integrity of the MCDU and that disable the MCDU for serious malfunctions.

Figure 6: FMS general architecture



As the FMS System is a part of the Auto Flight System (AFS) of the Airbus A330, the BITE Testing of the entire AFS System will be explained:

AUTO FLIGHT SYSTEM (AFS) BUILT-IN TEST EQUIPMENT (BITE)

Line Maintenance:

The line maintenance of the Automatic Flight System (AFS) is based on the use of the Fault Isolation and *Detection* System (FIDS).

The system:

- detects, isolates and memorizes the AFS internal and external faults
- initiates and performs the test after replacement of an AFS Line Replaceable Unit (LRU)
- initiates and performs the availability test of the category III automatic landing function.

Characteristics of the AFS Maintenance System:

- A certain number of the AFS maintenance system characteristics are common to all the aircraft systems. These characteristics concern:
- the Built-In Test Equipment (BITE) operating principle;
- transmission of fault messages in normal mode
- transmission of maintenance data in menu mode.
- the operational use of the Multipurpose Control and Display Unit (MCDU) up to access to the AFS REPORT/TEST pages

Safety:

Special precautions were taken at the maintenance system design stage to ensure safety.

- At test *level* (LAND TEST, SYSTEM TEST)
- Each test request made via the MCDU is accepted only if certain conditions concerning the components which perform the test are met:
  - The FIDS only accepts the test request if its ground condition is met (NOSE GEAR PRESSED and ENGINES STOPPED).
  - The LRU's under test only accept the test request if their own ground conditions are met (NOSE GEAR PRESSED and ENGINES STOPPED).
- At interference *level* between the software which performs the test and the application software.
  - The test software is only authorized to read certain variables of the operational software. In no case can it write into the memories where these variables are stored.

The system comprises:

- A FIDS card physically located in each Flight Management, *Guidance* and Envelope Computer (FMGEC). Only the card located in the command side of the FMGEC1 is activated (by the SIDE 1 signal)
- the BITEs located in the various AFS computers:
  - FMGEC's 1 and 2 (FE CMD/MONG, FG CMD/MONG, FM)
  - Flight Control Unit (FCU) (in each of FCU channels)
  - MCDUs 1, 2 and 3.

FIDS Card

The FIDS card includes:

- a CPU (*Microprocessor* and *associated* circuits)
- a memory module which *contains* the application program
- ARINC input/output circuits
- *discrete* input/output circuits.

The FIDS serves as the SYSTEM BITE (maintenance data concentrator).

The FIDS is linked in *acquisition* and reception to the CMS and is connected to the BITEs of the various AFS computers.

It receives commands (on the label 227) from the CMC, interprets these commands and transfers them to the various BITEs concerned.

It can also request the BITE to give *complementary* information.

It receives malfunction reports (labels 350- 351) from BITEs, manages these reports and, if applicable, consolidates the BITE diagnosis and generates a fault message which is sent to the CMC (label 356).

BITE information of MCDUs, FM and FG parts is sent to the FIDS via a common RAM.

AFS BITE

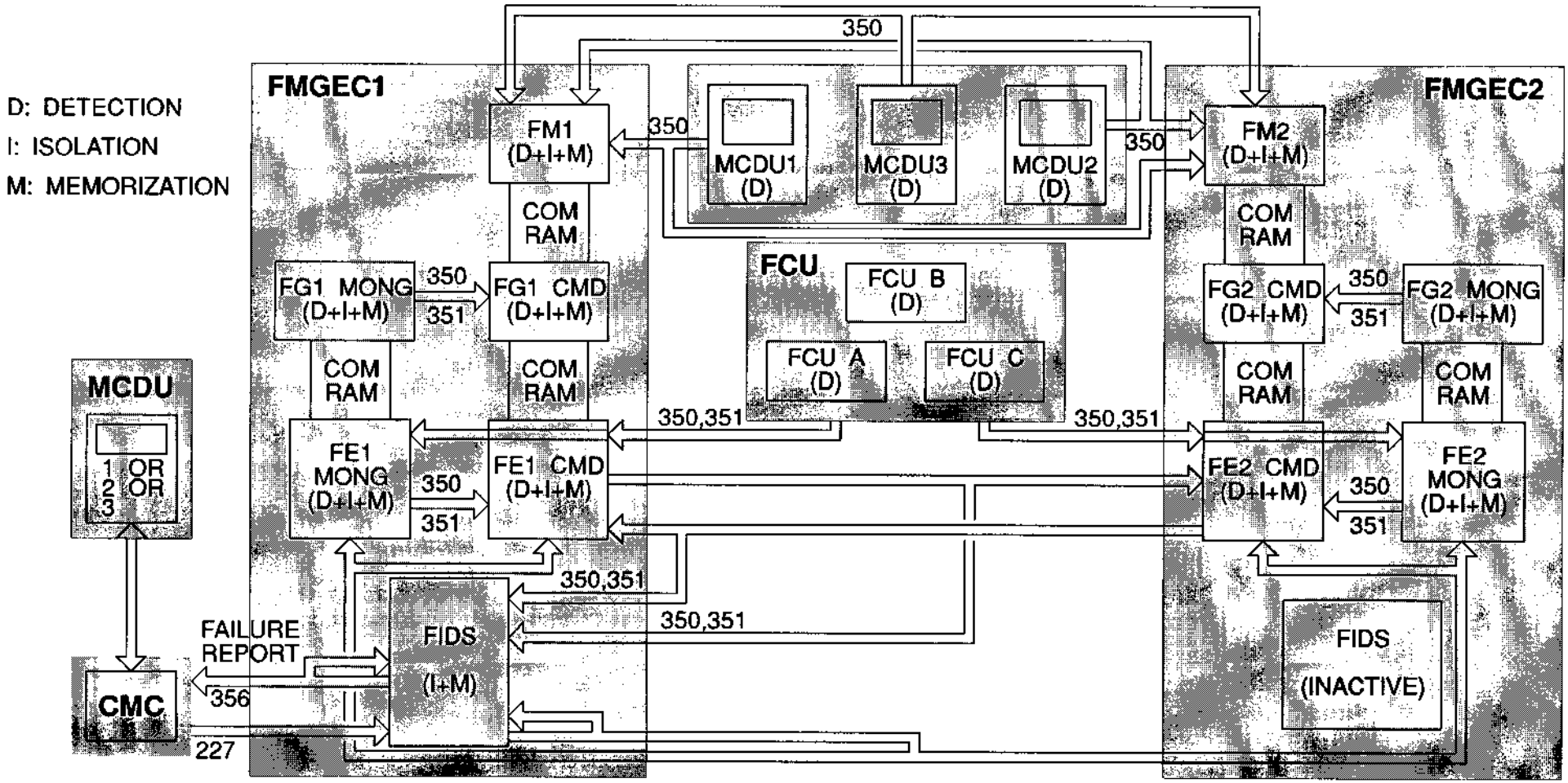
The BITE is an electronic device (HARDWARE + SOFTWARE) located inside each AFS computer.

Its main function is to detect, isolate and memorize the system failures.

At operating *level*, distinction must be made between:

FE and FG BITE, FM BITE, FCU BITE, MCDU BITE, FIDS BITE.

Figure 7: BITE and FIDS Block Diagram (AFS System)



Electronic Flight Control System (EFCS) Airbus A330

The electrical part of the flight control system *permits* the control of:

- one inboard aileron and one outboard aileron on each wing. Each aileron is actuated by 2 electro-hydraulic servo controls.
- 6 spoilers on each wing. Each spoiler is actuated by one electro hydraulic servo control.
- 2 elevators. Each elevator is actuated by 2 electro hydraulic servo controls.
- The Trimmable Horizontal Stabilizer, actuated by an actuator with two hydraulic motors and controlled by 3 electric motors.
- The actuator includes a device which makes it possible for the pilot to override the electrical control using the pitch trim mechanical control.
- The rudder, through the 2 yaw damper servo actuators. The actuator outputs are mechanically summed to the orders from the pedals to move the mechanical input of the 3 servo controls which actuate the surface.
- The rudder trim actuator, the travel limitation unit and the pedal limitation unit.
- The inboard aileron electro-hydraulic servo controls, the elevator electro-hydraulic servo controls and the yaw damper servo actuators are each connected to 2 computers: 1FCPC, 1FCSC. The other actuators are connected to one FCPC or FCSC.

To realize the various laws and functions, the flight control computers *acquire* signals from: The side stick, speed brake lever and throttle lever potentiometers. The rudder pedal inductive sensors. The rudder trim control switch and RESET push button switch. The FAULT/OFF and side stick priority push button switches. The hydraulic pressure transmitters and switches. The accelerometers and rate gyro unit. The wheel tachometers.

Fly By Wire Philosophy

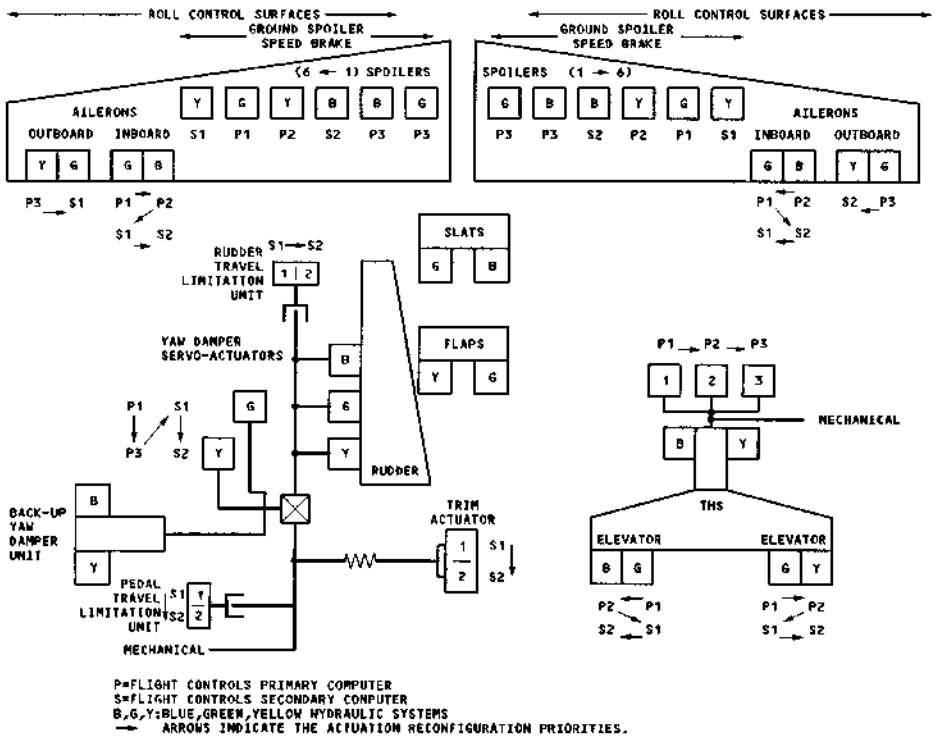
General

All the flight control surfaces are electrically controlled and hydraulically operated.

In addition, the stabilizer and the rudder have hydromechanical control back- up in case of electrical failure.

The main advantages of this philosophy are an *improvement* in redundancy, in handling qualities and thus in performances.

Figure 8: Electronic Flight Control System



Control Column

The control column has been replaced by the side stick.

Mechanical Linkage

The mechanical channel is replaced by electrical wiring and computers.

These computers control the servo- actuators.

The computers *elaborate* the flight control laws, including flight envelope protection, which optimize the control of the aircraft

Autopilot Servo

The autopilot commands are directly transmitted to the computers.

Artificial Feel

The modulated artificial feel is replaced by the side stick centering spring.

Servo Actuator

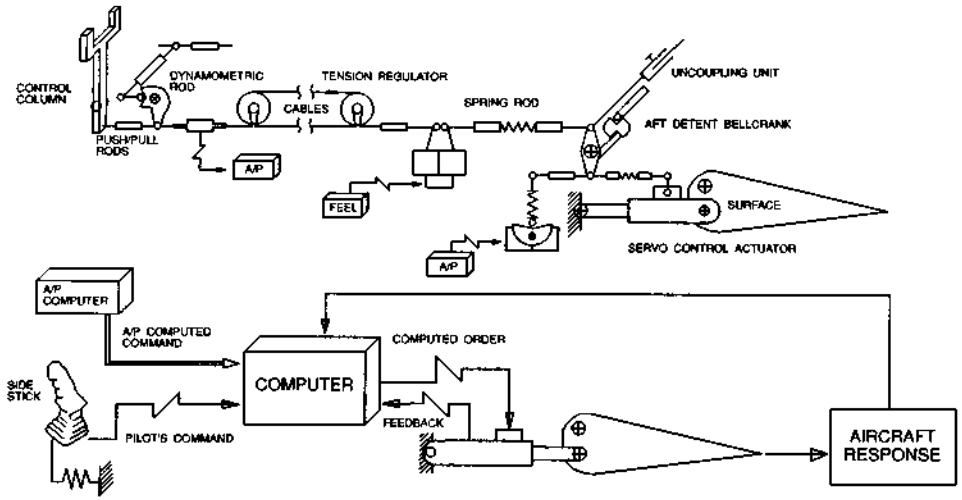
The mechanical feedback of the conventional servoactuator is replaced by an electrical feedback to the computers.

Control Wheel Steering Function

The Control Wheel Steering (CWS) function is ensured by the computers which use the aircraft response to maintain the required *attitude*.

Figure 9 shows the differences between the older "classic" Flight Control System and the modern Fly By Wire system.

Figure 9: Fly By Wire Philosophy



BITE Maintenance

The flight controls maintenance and failure *detection* function is built around the 2 Flight Control Data Concentrators (FCDC's).

The failure *detection* is ensured by the Flight Control Primary Computers (FCPC's), Flight Control Secondary Computers (FCSC's) and FCDC's.

The FCPC's and FCSC's send failure information to the FCDC's which analyse, store them and send maintenance messages to the Central Maintenance Computers (CMC's).

The failure data are accessible on the Multipurpose Control and Display Unit (MCDU) in the form of maintenance messages.

Tests in Interactive Mode

The electrical flight control system *comprises* 5 tests initiated on the ground from the MCDU. These tests are managed by the FCDC's:

- System test
- Elevator servo control damping test
- Inboard aileron servo control damping test
- Outboard aileron servo control damping test
- Back-Up Yaw Damper test.

Automatic Tests

Two types of automatic tests are built in the flight control computers:

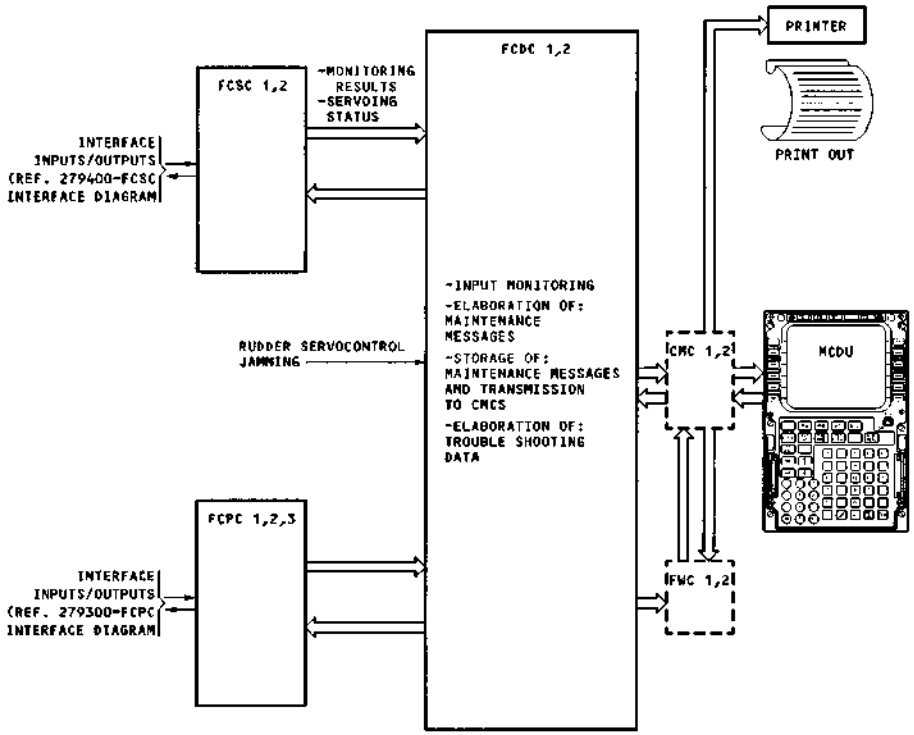
- the power-up tests
- the automatic tests performed at engine start to check availability of stand-by control channels.

Interface

- Interface between the FCPC's, FCSC's and FCDC's
- Interface with the Onboard Maintenance System (OMS)

The link between the CMC and the FCDC's is ensured by ARINC 429 buses; and is bi-directional.

Figure 10: Test Equipment of the EFCS System



Failure Detection

Except for the rudder servo control jamming *detection* which is ensured by the FCDC's, failure *detection* of the other LRU's is ensured by the FCPC's and FCSC's at the time of input validation by the monitoring related to servoing and the computer internal monitoring.

When a failure is detected:

- the FCPC's and FCSC's take operational actions (system reconfiguration) and send a failure information to the FCDC's.
- From this primary failure information (loss of servoing, etc.); the FCDC's *elaborate* a warning boolean and if necessary the *complementary* data required to animate the ECAM System page.
- Moreover, the FCDC's analyse the failure and, after confirmation, generate a maintenance message. The message gives in clear language the designation, FIN and ATA chapter related to the removal of the concerned LRU. This message is permanently transmitted to the CMC's.
- The FCDC's memorize the message and associate the time, date of failure occurrence and additional data linked to the failure.
- The memorization is performed in the FCDC 1 and 2 as per the rules below:
  - the system failures (ATA 27 except SLAT/FLAP system) are *considered* as internal failures
  - the others (e.g. ADIRU failures) are *considered* as external failures.
- The transmission is performed as per the rules below:
  - if both FCDC's are available, FCDC 1 sends external and internal failures while the FCDC 2 sends all failures as external to avoid double storage within the CMC.
- if the FCDC 1 is not available, the FCDC 2 sends failures classified as internal and external.

Global Positioning System (GPS)

Description and Operation

The Multi Mode Receiver (MMR) receives RF signals through an active GPS antenna (preamplifier *implemented* within the antenna). The GPS receiver filters, mixes, and performs analog-to-digital conversions. The resulting data is *processed* by *microprocessors* that output position, velocity, time, and integrity data to the system processor. The system processor transmits ARINC 743A - compliant data for use by other aircraft systems. The GPS receiver also outputs a time mark *discrete* signal that tells users of the ARINC 743A data the instant in time when the position solution is valid.

ADIRU

In normal operation, the MMR1 data is used by ADIRU 1 and 3; the MMR 2 data by ADIRU 2. In order to reduce GPS Receiver initialization time, ADIRU1 and 2 respectively send data to MMR 1 and 2 (IR position, *Altitude*, Date, UTC).

FMGEC

The *Inertial Reference* portions of ADIRU 1 and 2 respectively *provide* Flight Management *Guidance* and Envelope Computer (FMGEC) 1 and 2 with pure *inertial reference* data, *hybrid* GP *inertial reference* data used by the FMGEC for position fixing purposes and pure GPS data which is displayed on MCDU; in this case the ADIRU operates as a relay.

In case of failure of one GPS: the two ADIRU's automatically select the only operative GPS to compute *hybrid* GPIR data.

ATT/ HDG Switching

In case of failure of ADIRU 1, FMGEC 1 uses ADIRU 3 / GPS 1 data and in case of failure ADIRU 2, FMGEC 2 uses ADIRU 3 / GPS 2 data.

The primary source of ADIRU 3 being GPS 1, it is necessary to select the secondary input part of ADIRU 3 (GPS2) by means of the SWITCHING ATT/ HDG selector switch to preserve side 1 /side 2 segregation (GPS1/ ADIRU1/ FMGEC1 and GPS2/ ADIRU3/ FMGEC2 architecture).

DMC/ FWC

The MMR's are monitored by the three ADIRU's using the status word send by each MMR and the BITE of the *inertial reference* portion.

In case of GPS failure, the NAVGPS 1(2) FAULT message is displayed on the lower part of the Engine/ Warning Display (EWD).

This message is accompanied by:

- activation of the MASTER CAUTION lights on the glare shield,
- *aural* warning: Single Chime (SC).

Indicating

The GPS data is displayed on the GPS Monitor Page of the MCDU through the FMGEC.

The displayed data are:

- GPS position (Lat/ Long)
- true *track*
- figure of merit (in meters)
- ground speed
- mode.

Antenna

The GPS antenna is a L- band preamplified antenna. The GPS antenna is designed to operate at 1575.42 MHz with a right hand circular polarization and to *provide* an omnidirectional upper hemispheric coverage.

CMC

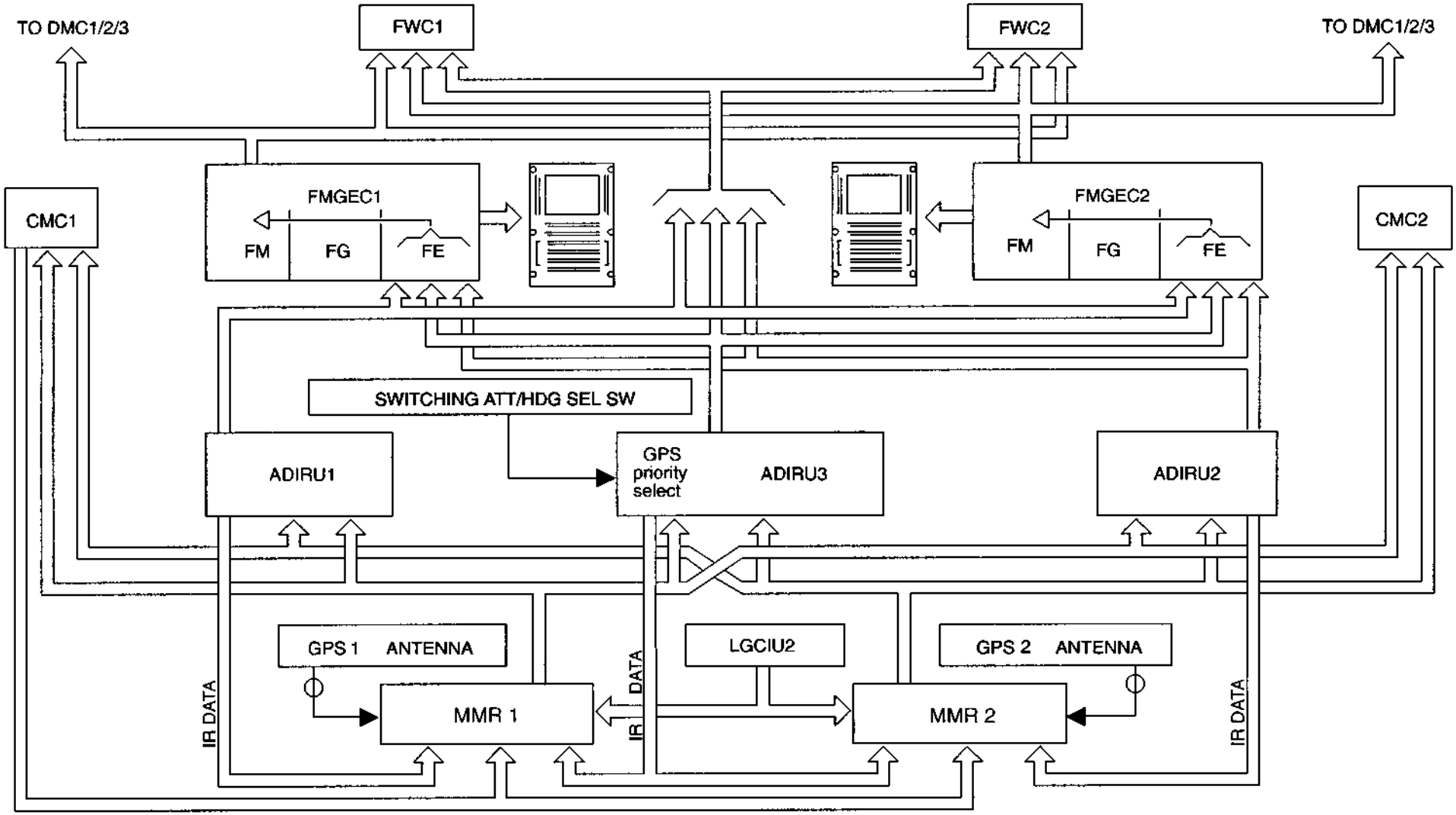
The BITE of the MMR is connected to the Central Maintenance Computer (CMC).

The units tested are the MMR, GPS Receiver, ILS Receiver, GPS antenna and coaxial cable.

LGCIU

This ground/ flight information is used by the receiver BITE module to count the flight legs.

Figure 11: GPS System Overview



T

U-Z

A

acceptable
-annehmbar

accommodate
-beherbergen, unterbringen

accomplished
-vollbringen, erreichen

accuracy
-Genauigkeit, Päzision

accurate
-genau, akkurat

accurately
-genau, präzise

achieved
-erreichen

adequate
-passend

advisory
-beratend, ermahnend

Amplifiers
-Verstärker

angular
-Winkel

ANSI
-American National Standards Institute, Amerikanischer Normenausschuss

applying
-anwenden, anlegen

approximation
-Annäherung, Angleichung

array
-Aufstellung, Ansammlung (comput)Datenfeld, eine Form des Datenaufbaus

assert
-festlegen, bestimmen

assigned
-zugewiesen, zugeteilt

assignment
-(comp) Zuweisung

assumption
-Annahme, Vermutung

at stake
-auf dem Spiel stehen

attenuation
-Verminderung, Abschwächung, Dämpfung

B

Barometric pressure
-Luftdruck

Base
-Basis

based
-basiert, auf Basis

beyond
-jenseits, dahinter

Bezel
-Fassung, Rahmen

bias
-(error), unsymmetric (comp) unsymmetrisch (Fehler), bias vt (electr, electron, techn) vorbelasten, unter Vorspannung setzen, vorspannen, vormagnetisieren

blind
-Blende

branch
-Zweig, Abzweigung

buffer
-Speicher, Pufferspeicher

buffered
-gespeichert

bug
-Index einer Instrumentenanzeige
-Wanze, Käfer, Ungeziefer, Bazillus, verstecktes Mikrofon (comput) Programmfehler oder Hardwarefehler, der zu einem Problem führt

C

cache
-Versteck, geheimes Lager, (comput) Speicherplatz, der Informationen enthält, die der Computer in einer kurzen Zeit braucht

capacitance
-Kapazität (elektr.)

carry
-(electron, chem) übertragen, mitführen, transfer (math) übertragen, mitführen

carry out
-durchführen

central processor unit
-zentrale Prozessoreinheit

circumference
-Umfang

circumstance
-Umstand

cladding
-Verkleidung, Hülle

coated
-beschichtet

comparator
-Vergleicher

compatibility
-Vereinbarkeit, Verträglichkeit

complement
-Komplement, Ergänzung, Vervollständigung

compound
-chemische Verbindung

conductive
-leitend

conductor
-Leiter

conjunction
-Verbindung, in Verbindung mit

consider
-Erwägen

considered
-in Betracht gezogen

contentions
-Auseinandersetzung, Kampf

convenience
-Bequemlichkeit, Komfort

convenient
-praktisch, günstig

convention
-Konvention, Abmachung

conventional
-konventionell

convert
-umwandeln

correspond
-entsprechen, korrespondieren

coupler
-(electr) Koppler, coupler connecting line (electr) Kupplungsleitung

CPU
-see central processing unit

cut-off
-unterbrechen, abschalten

cyan
-Zyan, Farbe zwischen blau und grün

D

Dagger
-Dolch, Schwert

decrement
-rückwärts zählen

density
-Dichte

derived
-abgeleitet

derives
-hervorbringen, streben nach

designator
-Kennzeichner, Benenner

despite
-trotz

determine
-bestimmen, festsetzen

development
-Entwicklung

differentiate
-differenzieren, unterscheiden

digit
-Ziffer, Finger

digitized
-digitalisiert

disabling
-ausschalten

distinguish
-unterscheiden

distortion
-Verzerrung

disturbing
-stören, unterbrechen

dotted
-getüpfelt, punktiert

drift
-Bewegung, Richtung,Tendenz

drop
-fallen, sinken

dye
-Farbstoff, Farbe

E

edge
-Ecke, Kante, Flanke

edge-triggered
-flankengetriggert

egress
-Ausgang, Verlust

embedded
-eingebettet

employ
-einsetzen, anwenden, beschäftigen

employed
-angewendet, eingesetzt

encoding
-Kodierung

endoscopy
-Endoskopie (Medizinische innere Untersuchung, Untersuchung von Körperhöhlen mit dem Endoskop)

enhanced
-verbessern, erhöhen, steigern

equivalencies
-Gleichwertigkeit

equivalent
-gleichwertig,equivalent

established
-erreicht, gesichert, gefestigt

etch
-ätzen, kupferstechen, radieren, schneiden, kratzen

eventually
-schliesslich, endlich

excessive
-übermässig

F

Fetch
-abholen

fetch
-versetzen

flip-flop
-Kippschaltung, Kippglied

floating
-(electr) erdfrei, nicht geerdet

floating-point
-Gleitpunkt, Gleitkomma

foundation
-Fundament, Basis

friction
-Reibung

G

Gain
-Gewinn, Vorteil, Zunahme, Steigerung

gain
-erreichen, gewinnen, zufügen

gap
-Lücke, Spalt, Abstand, Entfernung

generic
-generell, typisch

glance
-Blick, Glanz

glitch
-kleiner Fehler, (electr) Spannungsspitze, Störspitze

H

Hence
-daher, deswegen

hybrid
-Mischform, Kreuzung

I

IC
-integrated circuit
-Integrated Computer (Embraer)

IEEE
-Institute of Electric and Electronic Engineers, Institut für Elektrizitäts- und Elektronikingenieure

impedance
-Impedanz, elektrischer Scheinwiderstand

imperceptible
-nicht wahrnehmbar,unfassbar, unmerklich

implement
-realisieren, ausführen, erfüllen, durchführen

implementation
-Realisierung, Ausführung

impurities
-Verunreinigungen, Unsauberkeit

incandescent
-glänzend, glühend

incorporating
-enthalten, vereinigen

increment
-Erhöhung, Zuwachs

inductance
-Induktanz (elektr.) rein induktiver Widerstand

inductor
-Induktor, Transformator

Inertial Reference System
-Trägheitsnavigation

infallibly
-unfehlbar

ingress
-Zugang, Eintritt

instead
-anstatt

integrated-circuit
-integrierter Schaltkreis

integrating
-integrieren, ergänzen, zufügen

interrelated
-zusammenhängend

inversion
-Inversion, Umkehrung

J

justify
-rechtfertigen

L

Ladder
-Leiter

Lamination
-(electr) Ankerblech, -schnitt

laser
-(light amplification by stimulated emission of radiation) Gerät zur Steigerung der Lichtkraft mittels Strahlen, Strahlenwerfer

latency
-Latenz, (comput) die Zeit zwischen der Befehlseingabe und der Ausführung des Befehles

layer
-Ebene, Schicht

lend
-helfen

level
-Stufe, Niveau, Höhe, Ebene

loop
-Schleife,Schlaufe

loss
-Verlust

LSB
-least significant bit
kleinstwertiges Bit

M

magnitudes
-Grösse

margin
-Grenze, Rand, Spielraum

mauve
-Malvenfarbig

merely
-nur, lediglich

microprocessor
-Mikroprozessor

mitigating
-mildern, lindern

mitigation
-Milderung, Linderung

MSB
-most significant bit, höchstwertigest Bit

multiplex
-Uebertragungskanäle bündeln oder vielfach ausnützen

N

negation
-Negation, Verneinung, Ablehnung

Nibble
-halbes Byte, Dateneinheit mit 4 Bits

noise
-(electron) Rauschen, Störschall, Störgeräusch, Geräusche

Nought
-Null

O

occur
-auftreten, sich ereignen

occurring
-auftretend

odometer
-Kilometerzähler

opaque
-lichtundurchlässig, undurchsichtig

P

Parity
-Parität, Gleichheit

Peripherals
-Rand

perish
-vergehen, umkommen, zugrunde gehen

pictorial
-bildhaft, illustriert

pit
-Grube, Vertiefung

pixel
-Pixel, (comput) kleinste Dateneinheit die einen Punkt in einem Objekt in einer computerisierten Grafik darstellt

power of
-(Math.) Potenz, 3 to the power of 4, 3 hoch 4

precedence
-Vorrang, Priorität

predefined
-vordefiniert

predictable
-voraussagbar, voraussehbar

prefixed
-voranstellen

prevalent
-geläufig, gängig, verbreitet

previous
-vorhergehend

proliferation
-Wucherung, starke Vermehrung

promote
-fördern

propagate
-sich ausbreiten, verbreiten, vermehren

propagation
-Ausbreitung, Uebertragung

propagation delay
-Ausbreitungsverzögerung

punched
-gestanzt, gelocht

Q

quantisation levels
-Quantifizierungs-Stufen

queue
-Schlange, Warteschlange, Reihe

R

Radix
-Grundzahl

raising to a power
-hochrechnen

Random
-Zufall, zufällig

randomly
-beiläufig, zufällig

Range
-Bereich, Reichweite

ray
-Strahl

reciprocal
-reziprok, umgekehrt

recognise
-identifizieren, erkennen

recognition
-Anerkennung

recognize
-erkennen, identifizieren

reconfigure
-wiederherstellen, neu konfigurieren

redundant
-überzählig, überflüssig, doppelt ausgeführt

refractive
-Brechungs

relation
-Beziehung

relationship
-Verbindung, Verhältnis

reliance
-Verlass, Vertrauen

remainder
-Rest, Restbetrag

requirement
-Bedingung

rigid
-hart, starr, versteinert

S

saturated
-gesättigt, gefüllt

saturation
-Sättigung

sawtooth
-Sägezahn

semiconductor
-Halbleiter, Semikonduktor

significant
-bedeutsam, wichtig

sinks
-Senke, Abfluss

slope
-Steilheit,Abhang, abfallen

solid state
-(electron) monolithisch, Festkörper

spraying
-zerstäuben, besprengen, spritzen

spur
-anspornen

spurious
-unecht, künstlich,

stack
-Stapel, Haufen, (comput) ein Bereich im Speicher an dem man Informationen speichern kann und sie in der umgekehrten Reihenfolge abrufen kann.

staircase
-Treppenhaus

state
-festlegen, ausdrücken, bestimmen
-Zustand, Status, Staat

strand
-(electr) Litze

stray
-(electr) Streuung, Streuverlust

string
-Schnur, Kordel, (Comput) Zeichenfolge die als eine Einheit bezeichnet wird

stringent
-streng, überzeugend

stub
-Abzweigung

subscript
-tiefgestellt

successive
-eins nach dem anderen, folgend

swap
-vertauschen, austauschen
-Wechsel, Tausch

T

tedious
-mühsam, ermüdend, langweilig
threshold
-Schwelle
trace
-verfolgen, Spur untersuchen
track
-Kurs über Grund
-Spur
-verfolgen, ausfindig machen
trade-off
-einen Kompromiss machen, Konzessionen einräumen.
transient
-Spannungsspitze
transition
-Uebergang, Veränderung
treasure
-Schatz
trigger
-auslösen, betätigen
triggered
-ausgelöst, betätigt
tristate
-Dreistufen
tungsten
-Wolfram

U

unique
-einzigartig
unpredictable
-unvorhersagbar, unvorhersehbar

V

vary
-verändern, variieren
vast
-gross, enorm
versus
-gegen, gegenüber
violate
-brechen, verstossen, verletzen

W

waveguide
-Wellenleiter
wavelength
-Wellenlänge
weight
-Gewicht, Schwere, Gewichtigkeit
weighted
-gewichtet, abgewogen
weighting
-gewichten
withstand
-widerstehen, aushalten

Y

yarn
-Garn
yield
-hervorbringen, produzieren, Ertrag

yoke
-Joch, Paar, Verbindung, (elektr) Ableitungsspule, Elektromagnet der zur Richtung von Elektronenstrahlen in Monitoren dient