AVDASI 1
AENG 10001

Computing and data buses

University of BRISTOL

DEPARTMENT OF aerospace engineering

Content

- Aircraft Computing
  - Uses, types and development
  - Generic architecture and Characteristics
  - Instructing a computer
  - PIC example

- Data Buses
  - 1553
    - Hardware
    - Message types
  - AFDX
  - Federated and modular avionics

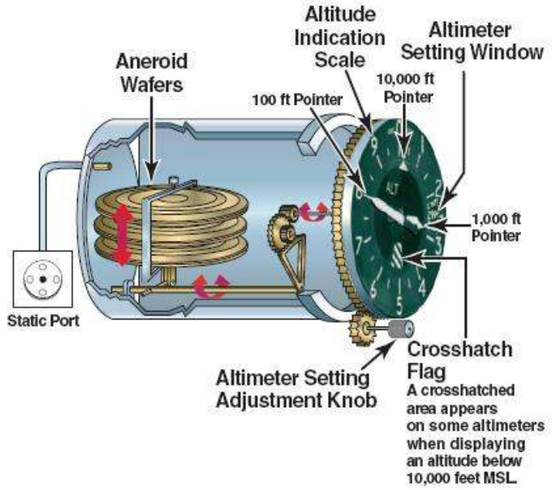
Aircraft Computing

- Current aircraft use substantial computing power for a variety of functions, some of which we have seen;
  - Signal processing
  - Displays
  - Navigation, Communication, Flight management, etc.

- Developed from;
  - Mechanical – ‘puff & blow’ air data *(extremely limited functionality)*
  - Analogue – linear circuits: op-amps, transistors *(liable to drift)*
  - Digital – integrated circuits *(improved performance, reduced size etc)*

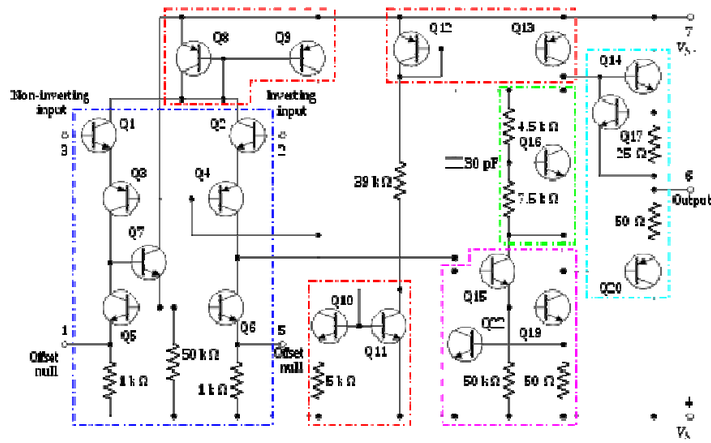
Technologies

Mechanical



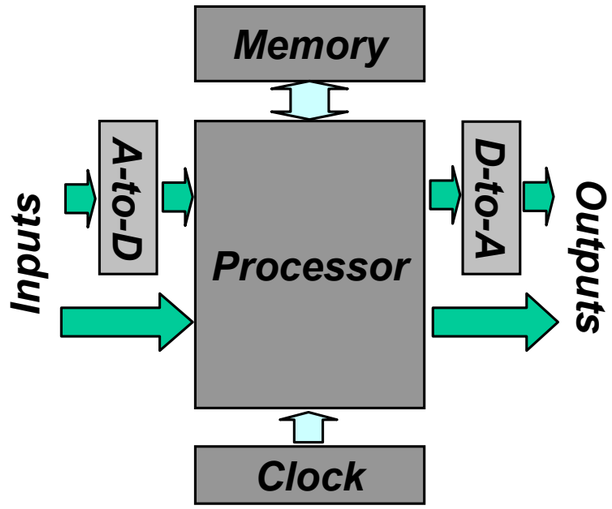
- Limited functionality
- Mechanically complex
- Low power

Analogue



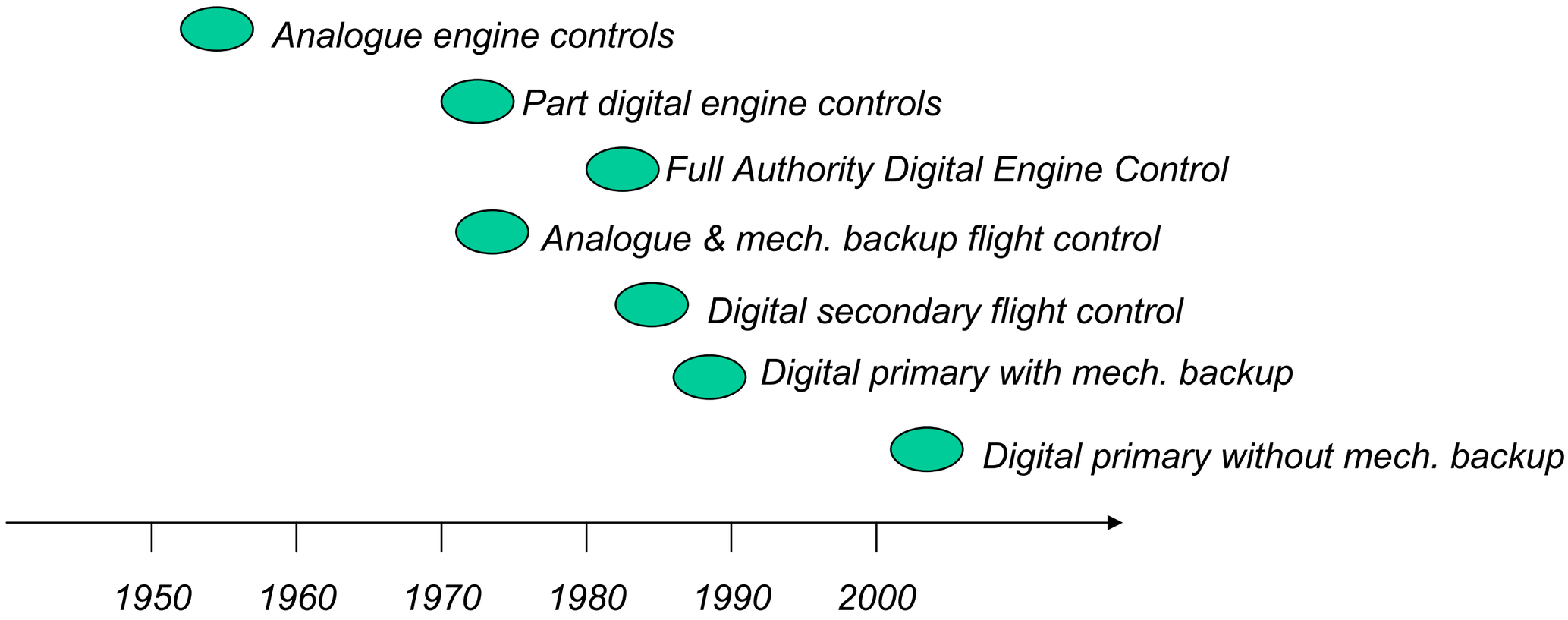
- Greater functionality
- High bandwidth/ fast response
- Liable to drift with time/temp
- Susceptible to noise

Digital



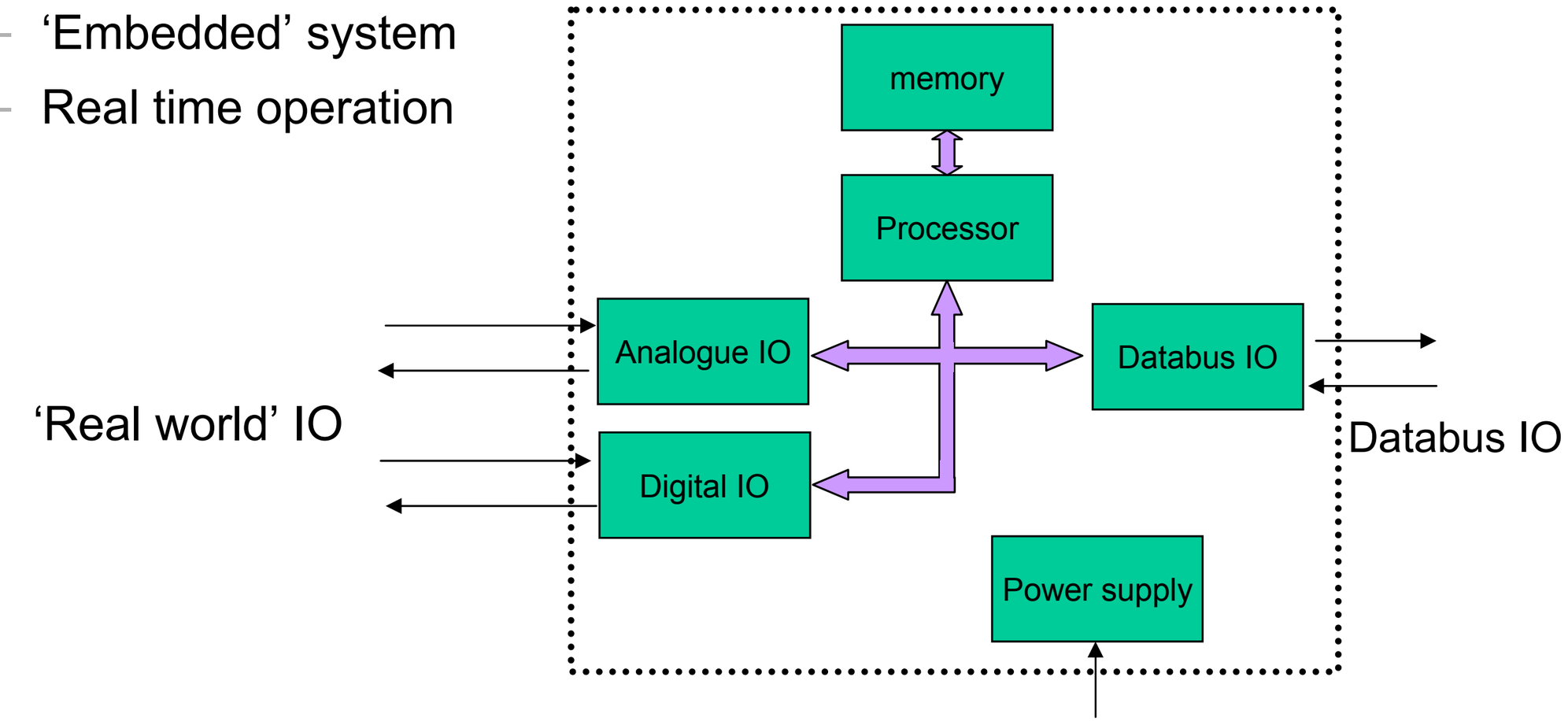
- Greatest functionality
- Greatest complexity
- High bandwidth/ fast response with modern architectures
- Resistant to drift
- Less susceptible to noise
- High power

Computing in engine and flight control



Generic airborne computer

- Key characteristics:
  - Task oriented computer
  - ‘Embedded’ system
  - Real time operation



May appear to be much ‘lower spec’ than your laptop……

Characteristics

- Task–oriented
  - Designed to do one or few specific functions

    e.g. drive a display or process a sensor output

- Embedded system
  - Part of a wider system

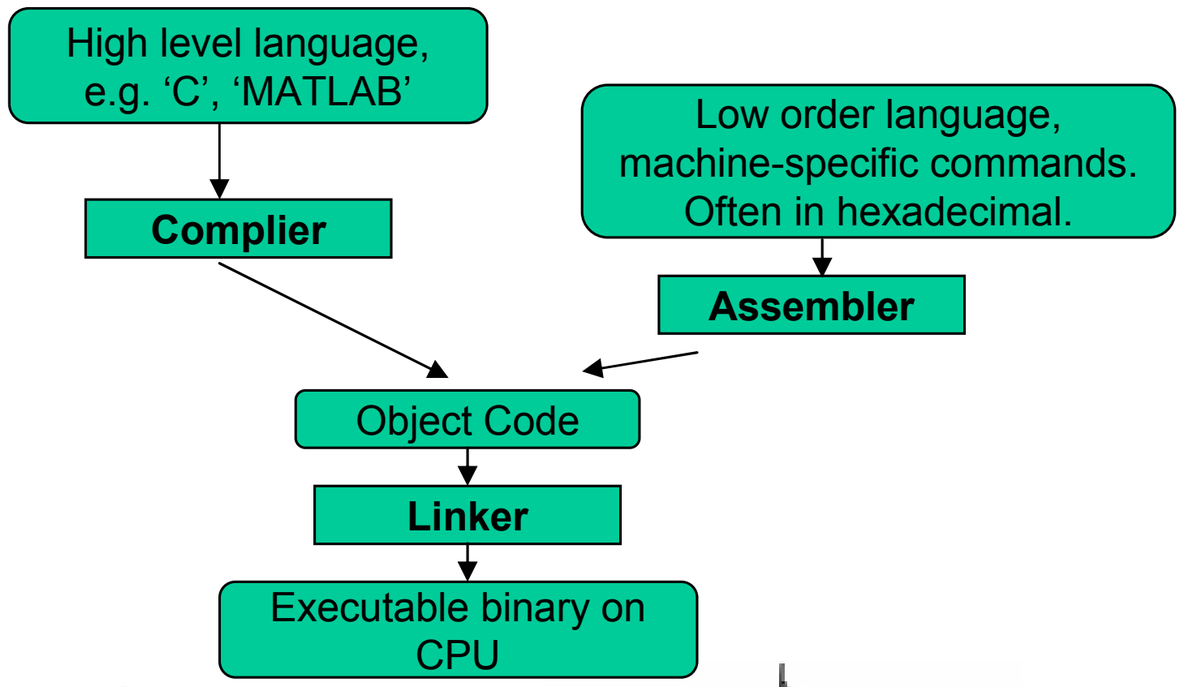
    e.g. part of the flight control or navigation systems

- Real Time
  - Able to perform task within a defined timescale.

    Essential for control and operation of an aircraft. Requires particular processor architecture and operational design.

Instructing a computer

- There are several differing hardware architectures that can be used to implement the CPU e.g. PowerPC, PIC, Intel 8086.
- The program information is contained in non-volatile memory as binary sequences (machine code).
- Since it is difficult to work directly with machine code, higher level languages have been developed for programming.



Example low-level code – PIC Assembler

```
Main:
        bsf     STATUS,RP0          ; Select Registers at Bank
        movlw   0x70                ;
        movwf   OSCCON              ; Set the internal clock speed to 8 Mhz
        clrf    TRISIO              ; Set all General Purpose I/O to output, TRISIO<3> always reads '1', always an input.
        clrf    ANSEL               ; Make all ports as digital I/O
        bcf     STATUS,RP0          ; Back to Registers at Bank 0

MainLoop:
        bsf     GPIO, b'0000100'    ; Set GP2 Output To High
        call    Delay               ; Call Delay Subroutine
        bcf     GPIO, b'0000100'    ; Clear GP2 Output To Low
        call    Delay               ; Call Delay Subroutine
        nop                         ; No Operation
        goto    MainLoop            ; Goto MainLoop

Delay:
        movlw    0xFF               ; Move literal 0xFF to working register
        movwf   Delay2              ; Move working register to register file

DelayLoop1:
        movlw   0xFF                ; Move literal 0xFF to working register
        movwf   Delay1              ; Move working register to register file

DelayLoop2:
        decfsz  Delay1,f            ; Decrease Delay1, If zero skip the next instruction
        goto    DelayLoop2          ; Not zero goto DelayLoop2
        decfsz  Delay2,f            ; Decrease Delay2, If zero skip the next instruction
        goto    DelayLoop1          ; Not zero goto DelayLoop1
        return                      ; Return to the Caller

        end
```

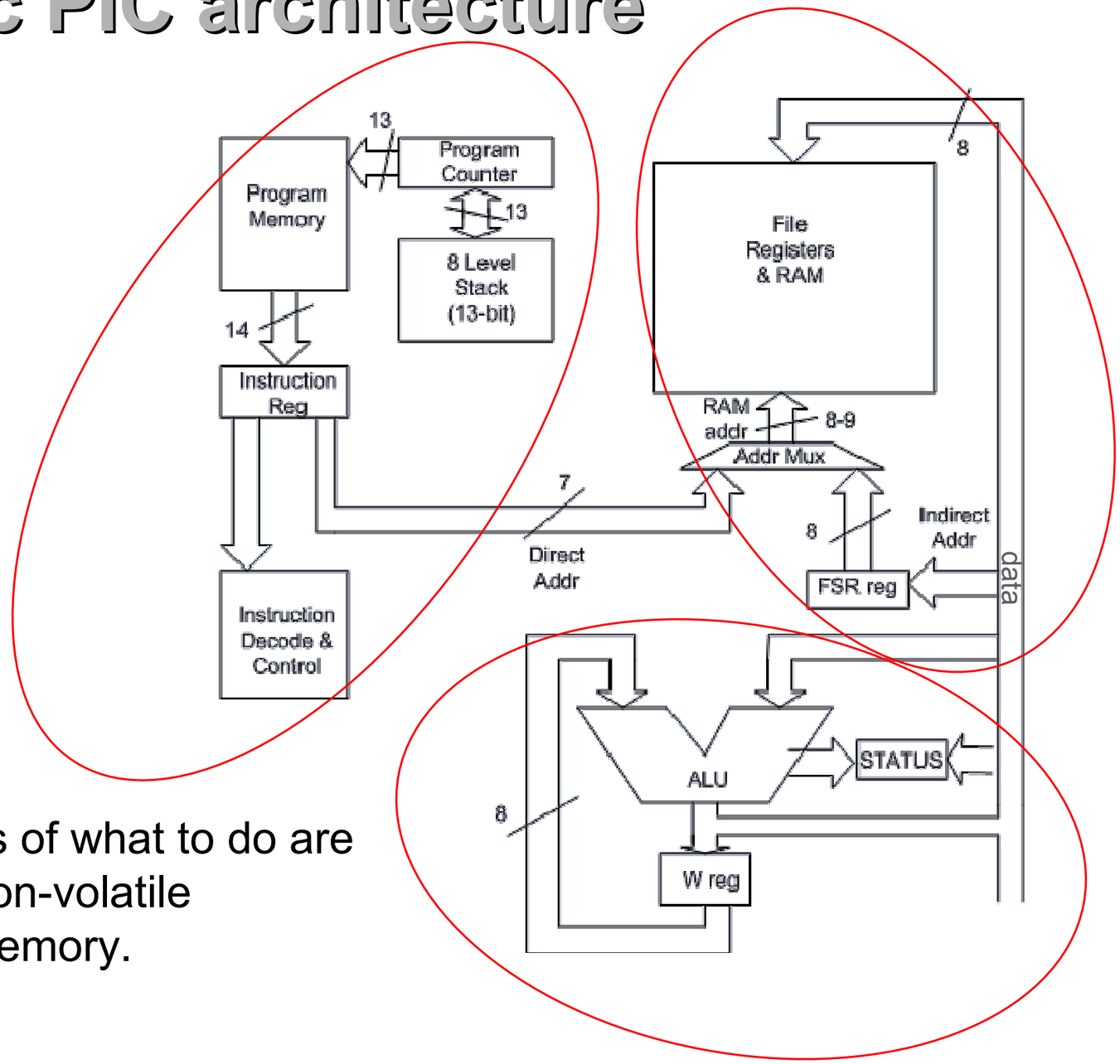
Operations

Operands

Routines

Comments (Not part of code)

Basic PIC architecture



Registers can be set to different values to configure inputs and outputs etc. or to actually output data. RAM can store working data.

Instructions of what to do are stored in non-volatile program memory.

The ALU, or Arithmetic Logic Unit, does the sums and returns the value in the W, or working, register.

University of BRISTOL

AVDASI

DEPARTMENT OF aerospace engineering

Example low-level code – PIC Assembler

```
Main:
        bsf     STATUS,RP0          ; Select Registers at Bank
        movlw   0x70                ;
        movwf   OSCCON              ; Set the internal clock speed to 8 Mhz
        clrf    TRISIO              ; Set all General Purpose I/O to output, TRISIO<3> always reads '1', always an input.
        clrf    ANSEL               ; Make all ports as digital I/O
        bcf     STATUS,RP0          ; Back to Registers at Bank 0

MainLoop:
        bsf     GPIO, b'0000100'    ; Set GP2 Output To High
        call    Delay               ; Call Delay Subroutine
        bcf     GPIO, b'0000100'    ; Clear GP2 Output To Low
        call    Delay               ; Call Delay Subroutine
        nop                         ; No Operation
        goto    MainLoop            ; Goto MainLoop

Delay:
        movlw    0xFF               ; Move literal 0xFF to working register
        movwf   Delay2              ; Move working register to register file

DelayLoop1:
        movlw   0xFF                ; Move literal 0xFF to working register
        movwf   Delay1              ; Move working register to register file

DelayLoop2:
        decfsz  Delay1,f            ; Decrease Delay1, If zero skip the next instruction
        goto    DelayLoop2          ; Not zero goto DelayLoop2
        decfsz  Delay2,f            ; Decrease Delay2, If zero skip the next instruction
        goto    DelayLoop1          ; Not zero goto DelayLoop1
        return                      ; Return to the Caller

        end
```

"Clear register 'F' "

..where 'F' is the TRISIO register. Clearing it sets some pins on the device to be an output

Move literal 0xFF to working register

Move contents of W register to a register we have called 'delay2'

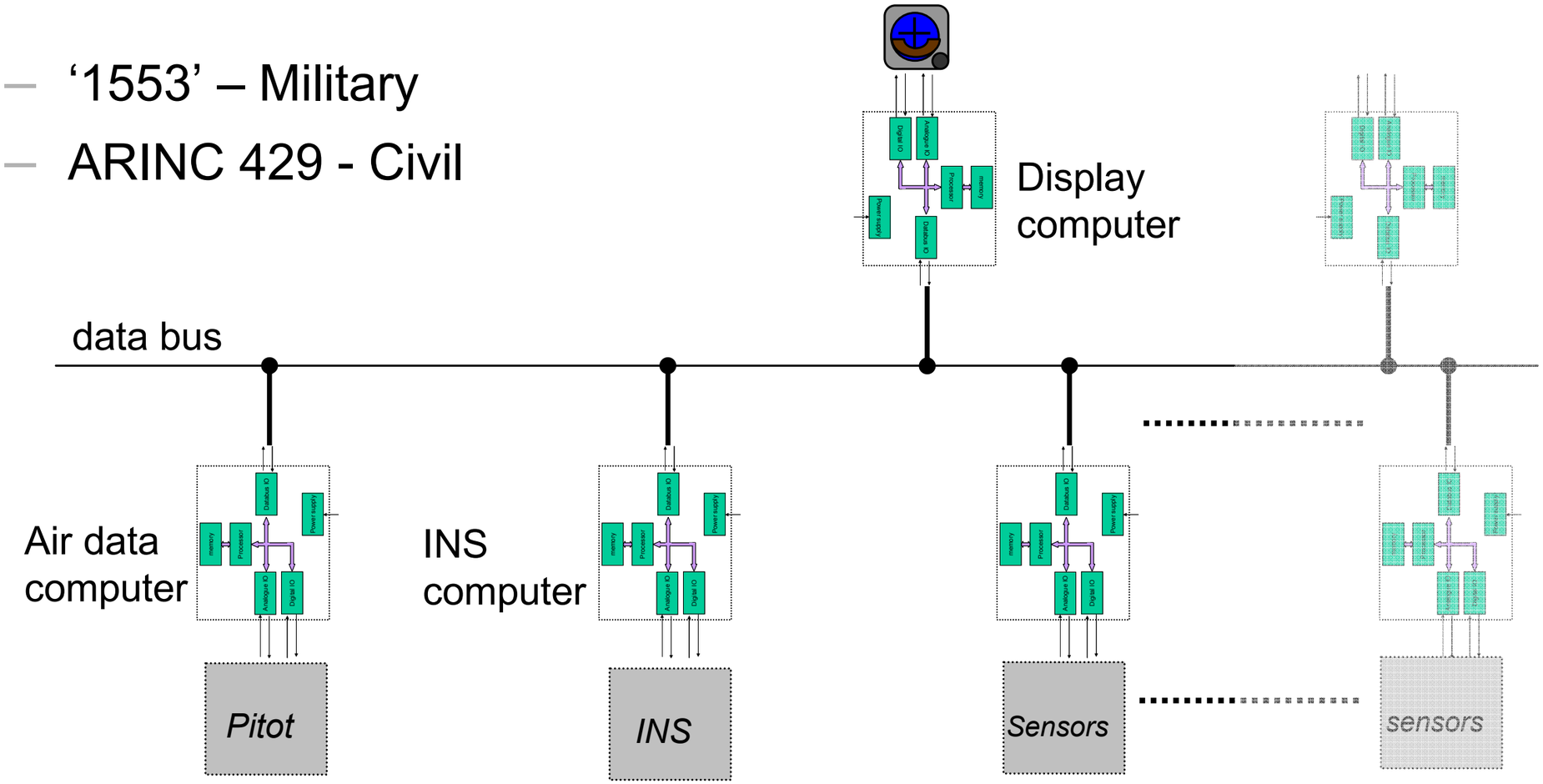
Aircraft Computing

- Initially aircraft computing closely followed new technological developments – maybe even led them.

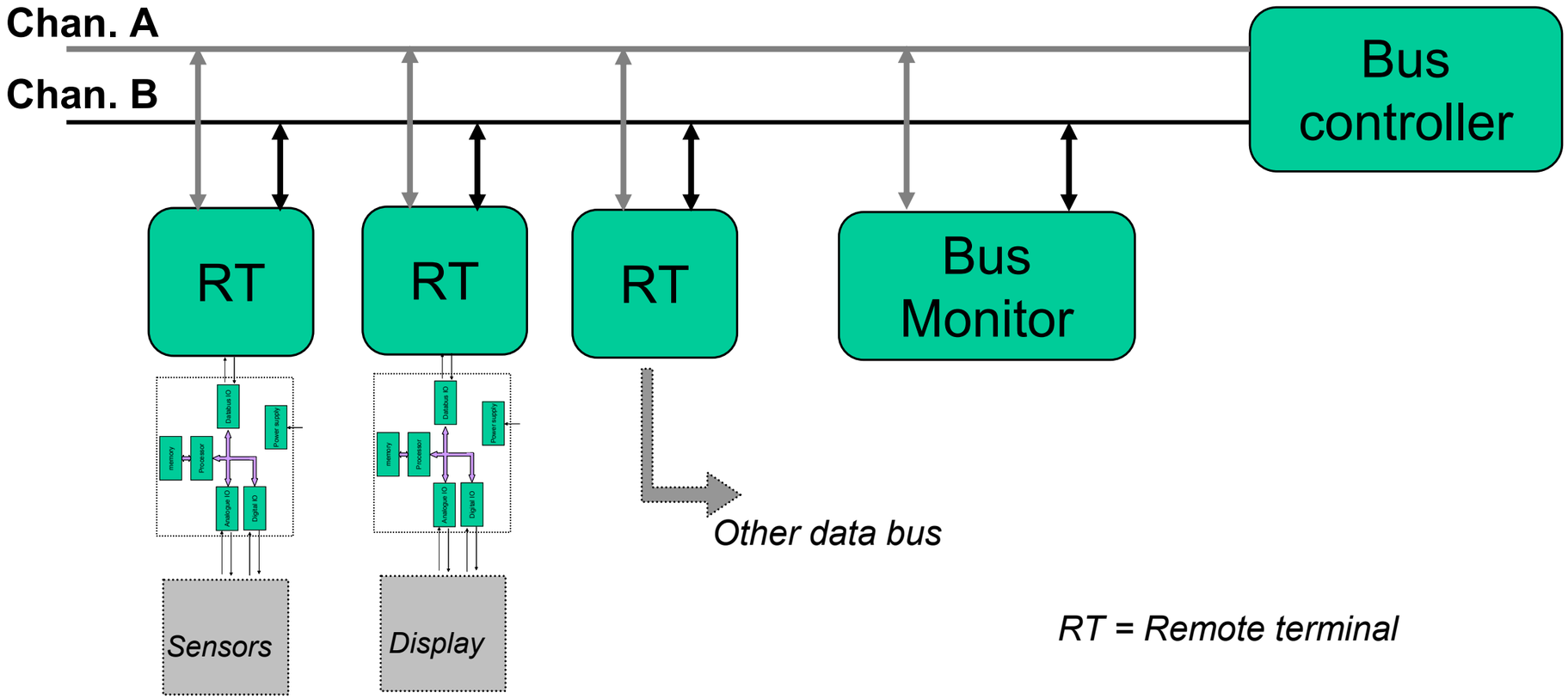
- However current technology development is dominated by high volume manufacture (like toys):
  - *Aerospace is characterised by relatively low volume, high performance and long lifetime – airframes last decades*
  - *Aerospace no longer drives technology and so has had to adapt*
  - *Concepts like 'COTS' (commercial off the shelf) have had to be embraced*
  - *But obsolescence is a major concern*

Data transfer - Buses

- The components of the avionics system are typically connected to each other by data buses.
  - ‘1553’ – Military
  - ARINC 429 - Civil



MIL-STD-1553B Bus



- A 1553 installation will typically feature a dual redundant structure

MIL-STD-1553B Bus

- Bus controller
  - Commands the various remote terminals to send or receive messages
  - Services requests from the remote terminals
  - Operates according to a commands stored in local memory.

- Remote terminal
  - Interfaces between data bus and external subsystem or another databus.

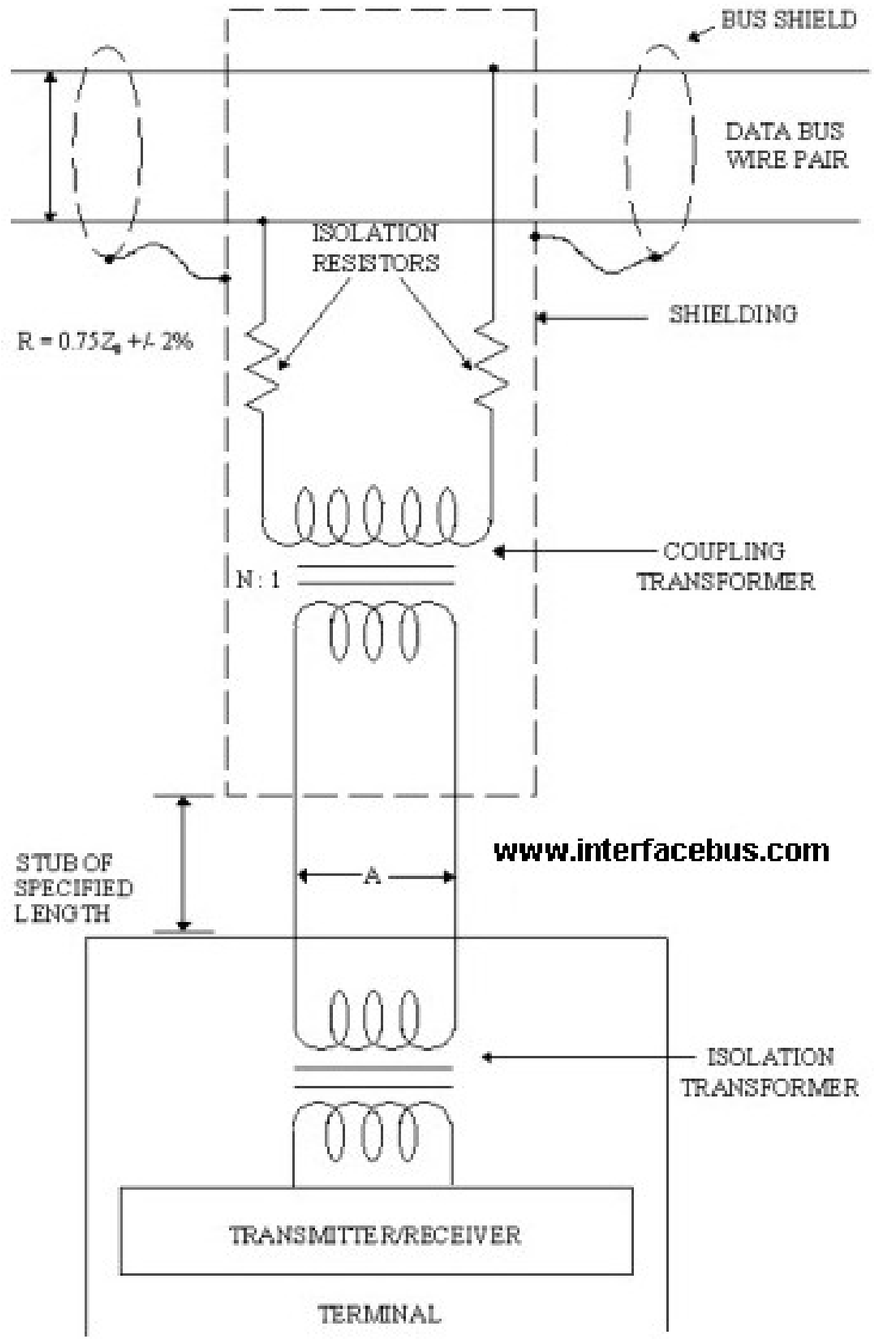
- Bus monitor
  - Monitors and records traffic over the bus for off-line analysis.

1553 hardware

1553 is a serial digital bus – data is sent one bit at a time.

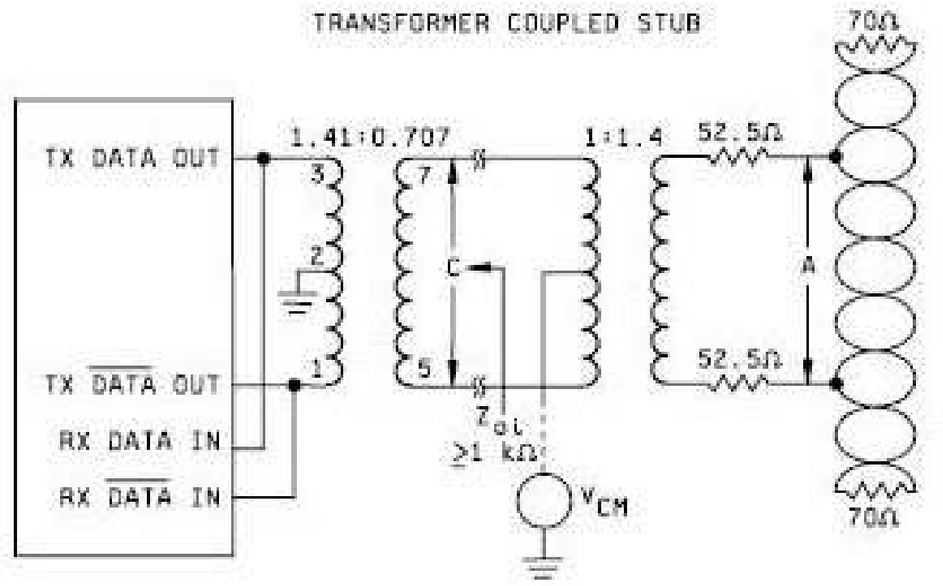
Each 1553 channel consists of a twisted wire pair. The pair are driven by digital signals in anti-phase

A digital ‘1’ is represented by a voltage level of approximately 20V (although the standard defines logic level ‘1’ as anything ~1V-20V)

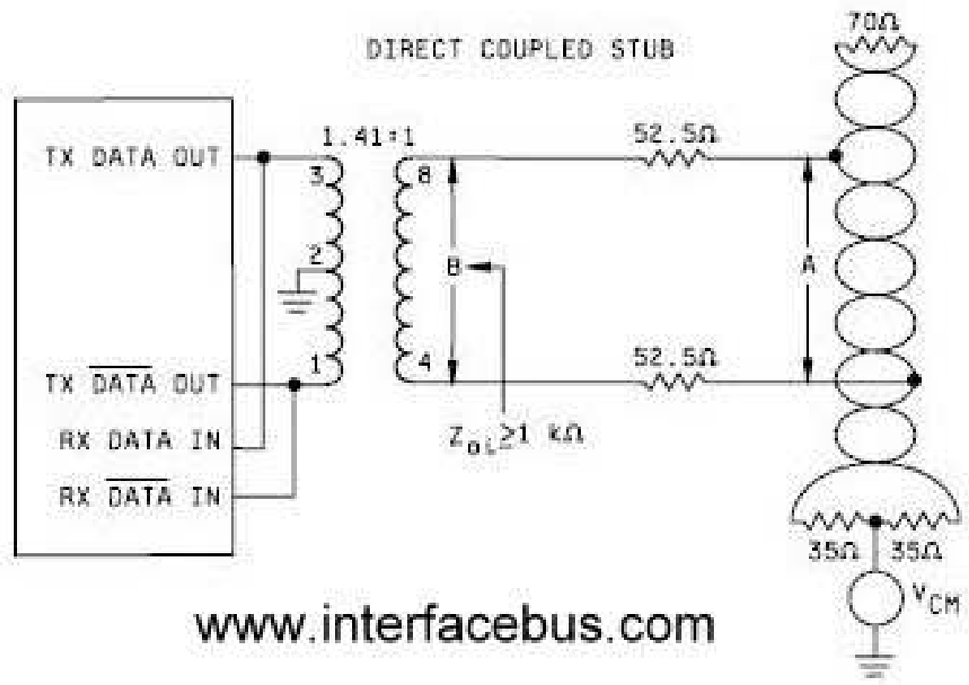


1553 hardware

- The transformer coupling to the bus makes the system more resilient to faults, particularly short circuits in a RT.



- RTs can be direct coupled or connected through a further isolation transformer. The isolation transformer allows the RT to be up to 6 metres from the bus (direct connection allows only 30cm).



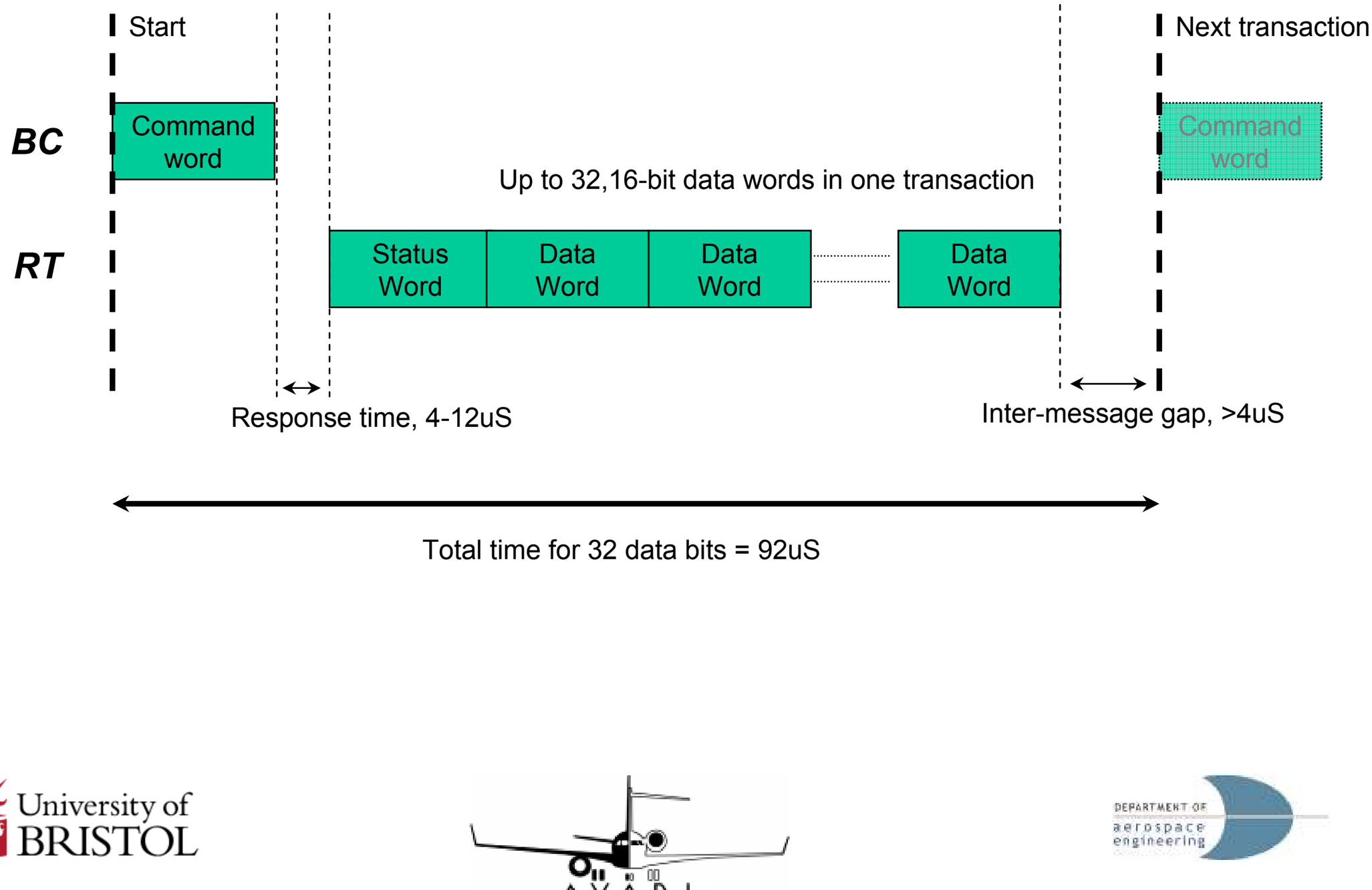
www.interfacebus.com

1553 words

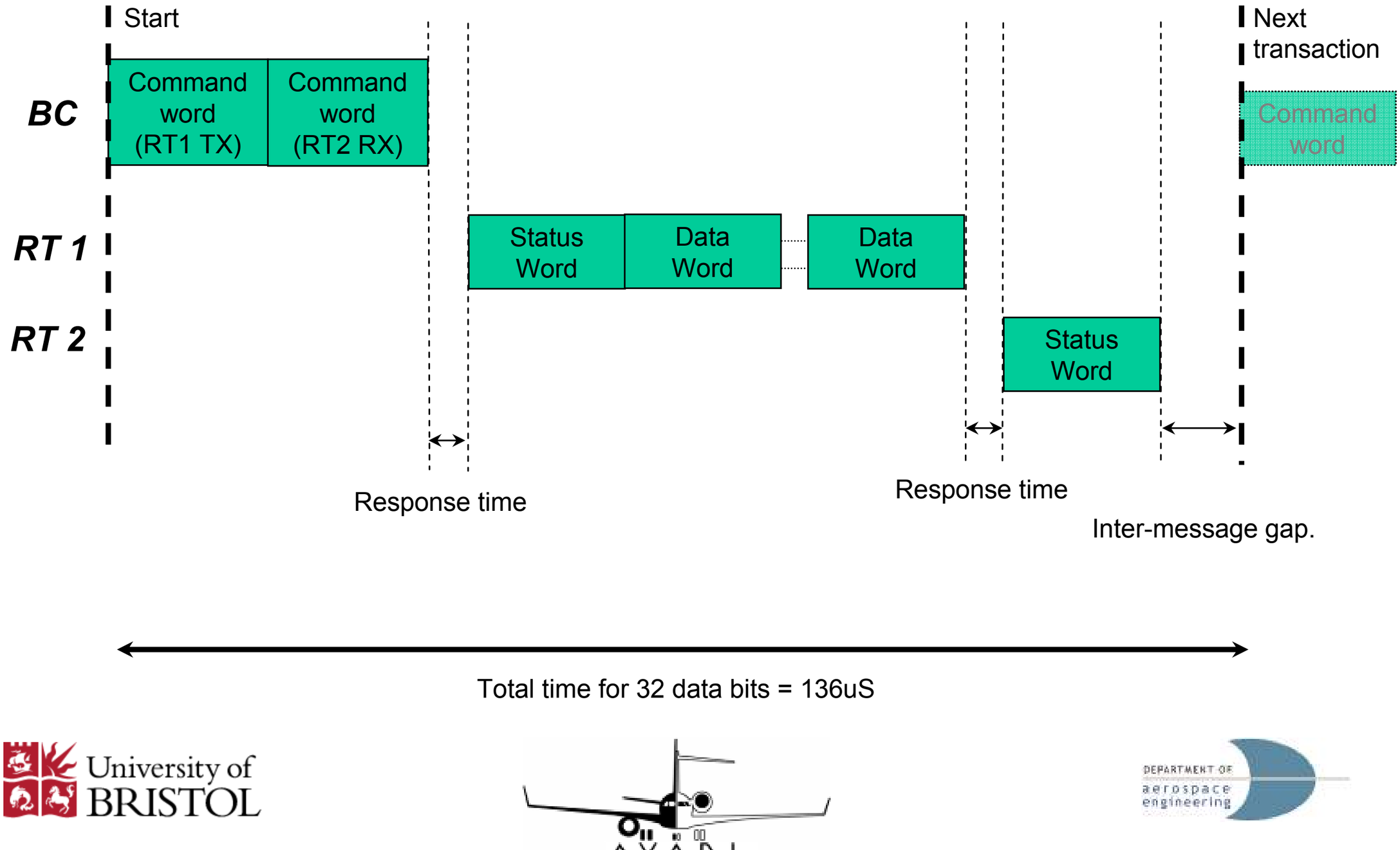
- The 1553 protocol consists of three type of word;
  - Data Word — *The information we wish to exchange (only 16 bits are actually data)*
  - Command Word — *Instruction from the bus controller*
  - Status Word — *response to the BC from the remote terminal*

- Each word is 20 bits and the transmission rate is 1Mb/s (=1uS per bit).
- There are 4 message formats;
  - BC to RT
  - RT to BC
  - RT to RT
  - Broadcast

RT to BC transaction

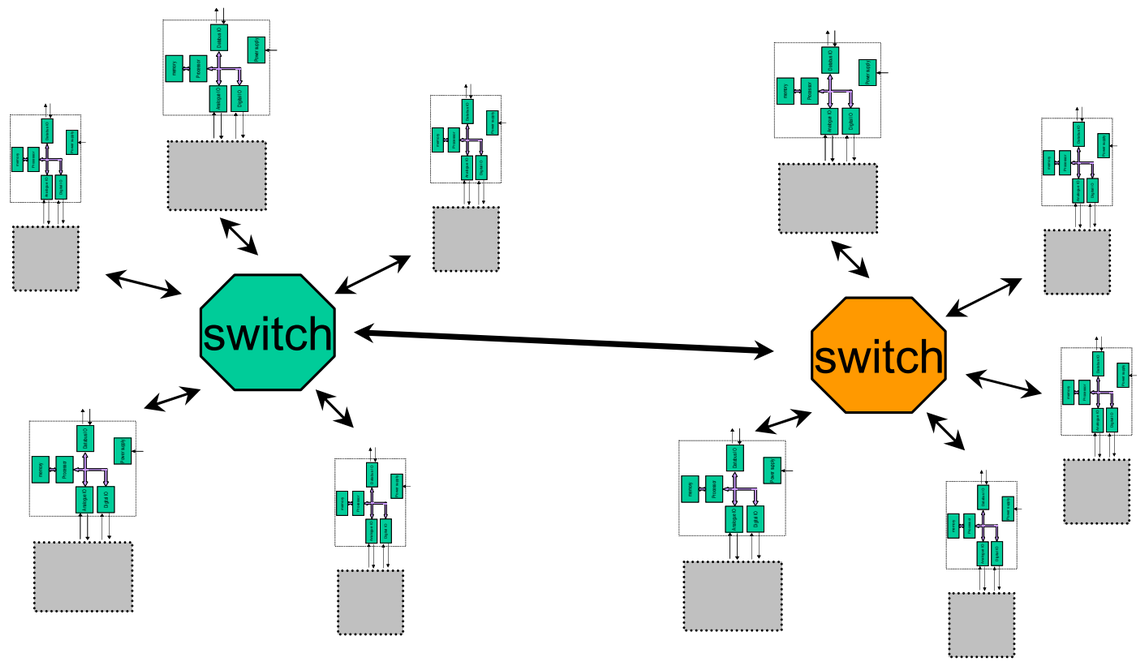


RT to RT transaction



ARINC 664 AFDX

- AFDX – Avionics Full-Duplex Switched Ethernet
  - A recent databus standard based on widely used IEEE 802.3 Ethernet standard.
  - AFDX employs the similar hardware as the commercial standard ethernet (although ruggedised), but with additional constraints in software to make the system more ***deterministic.***



AFDX arranges users in a star topologies with a switch at the hub. Multiple hubs can be connected together

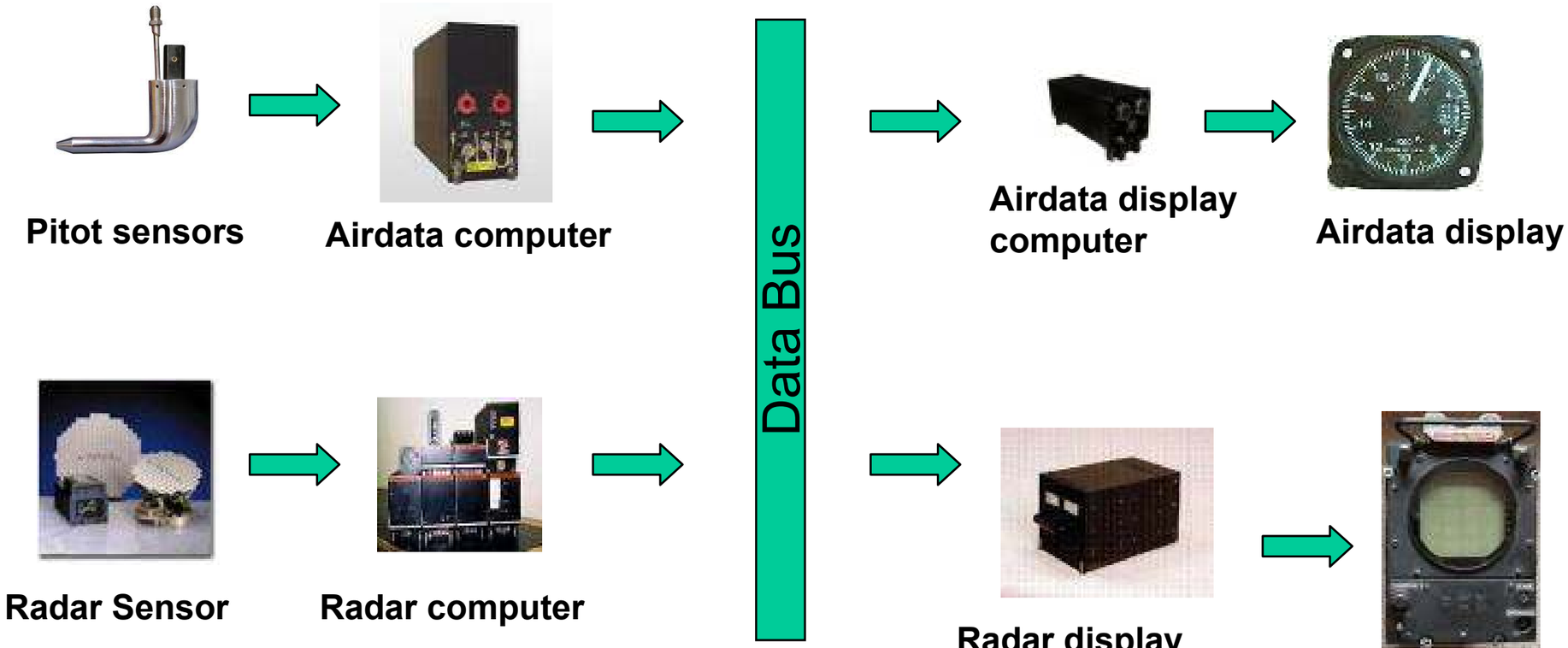
Avionic databus overview

- **Military**
  - MIL-STD-1553
    - US and NATO Military standard since 1973. 1Mb/s
  - MIL-STD-1773
    - Optical version of 1553
  - STANAG3910
    - Upgrade to 1553 used in Typhoon. Augments 1Mb/s bus with 20Mb/s high speed fibre optic bus.

- **Civil**
  - ARINC 429
    - Civil bus used for ~30 years on B737, B747, A330, A340 etc. 100 Kb/s
  - ARINC 629
    - Upgrade of 429 for B777, but not widely used. 2Mb/s
  - ARINC 664 AFDX
    - Latest high speed bus used on A380, B787, AW101 etc. >100Mb/s

System architecture – federated

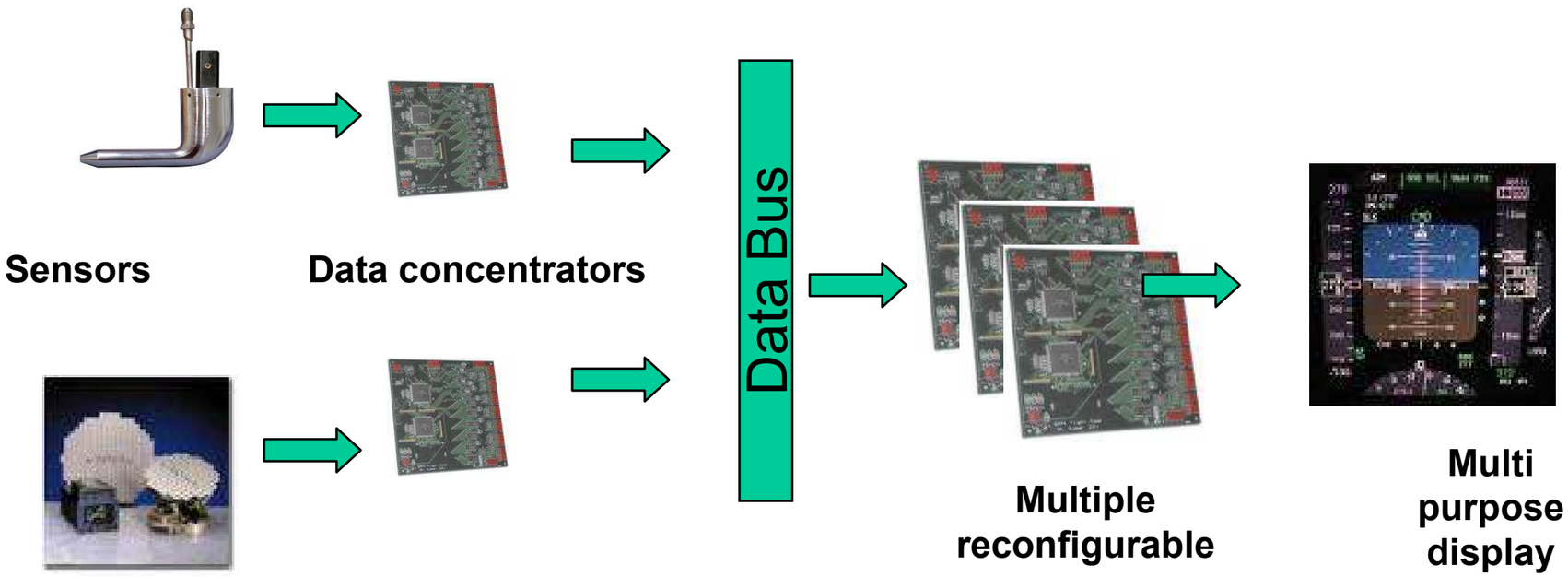
- Traditionally the system was designed with specialist units for each function – a 'federated architecture'



Federated Concept

System architecture – modular

- The most recent systems follow a ‘modular architecture’ where functions are shared.



Modular Concept

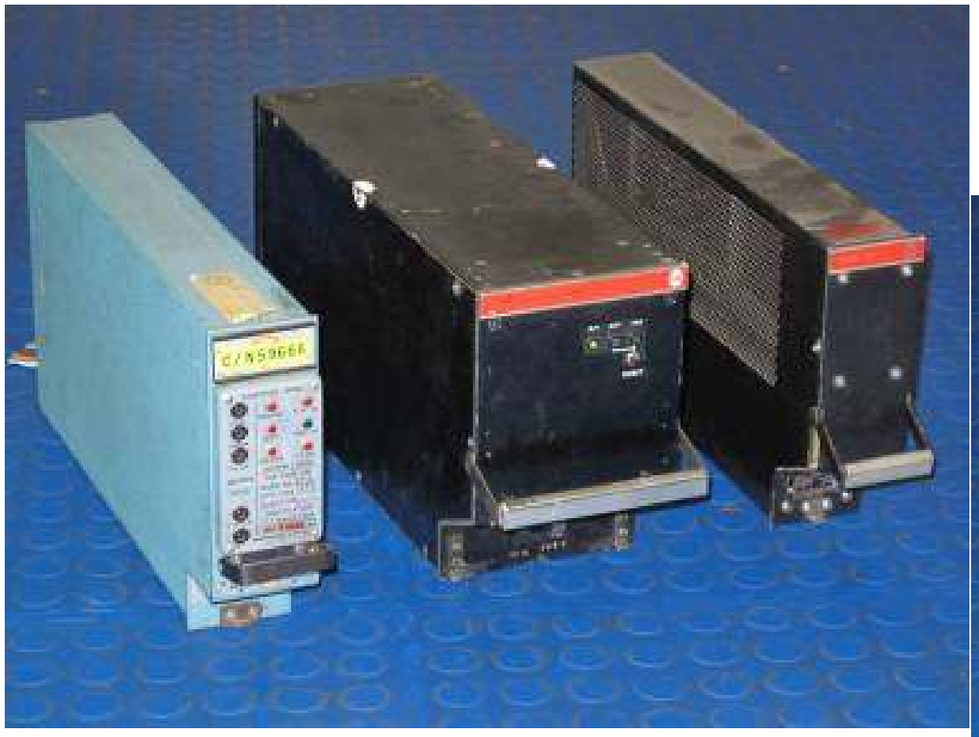
System architecture - LRU

- The various computers are housed in line replaceable units and sit in avionics bays. The connections are made via a back plane bus

LRUs in racking

System architecture - LRU

- The various computers are housed in line replaceable units and sit in avionics bays. The connections are made via a back plane bus



LRUs front

LRUs back