THE IMPLEMENTATION OF AN ON-BOARD DATA HANDLING SYSTEM BY A PROGRAMMABLE MODULAR CONCEPT

HORST FRÖLICH
DORNIER SYSTEM GMBH
Friedrichshafen, Federal Republic of Germany

Summary. The implementation of On-Board Data Handling Systems by individually tailored electronics leads to high development costs and risk. In order to overcome these shortcomings for future applications of on-board tasks, a complete modular concept has been developed for control, data acquisition, data reduction and telemetry encoding in onboard aerospace vehicles.

This highly flexible concept consisting of a large spectrum of real time peripherals and different central processing units has meanwhile proven its high flexibility in a wide spectrum of applications. The system is freely programmable and supported by extensive software for program development and testing. The application software can be developed on host computers of different types.

The paper gives a description of the system. By the example of an application in an aerospace telemetry and control system the advantage of using this modular approach is demonstrated.

1. Introduction. In recent years a certain evolution has taken place in the implementation of on-board data handling systems in the aerospace field. With the advent of microprocessors and large scale integrated circuits a new basis was given for more flexible system designs. Flexibility due to system programmability, which originally had an high impact on cost, as expensive on-board computers were required, is now feasible without that cost impact. New methods how to handle the engineering problems during the design of a data handling system were introduced. The implementation of a system, widely based on proven modular hardware, facilitates the engineering work and transfers the engineers workload to more challenging areas as system engineering, software development and system integration.

Before implementing a multi-purpose on-board data handling system as described herein, it is necessary to analyse common requirements of different types of applications. Modular concepts with inherent processing capability must be designed as a whole, i.e. the controlling unit and the real time peripherals must be integrated and clearly defined in one system concept. Clear electrical and mechanical interface definitions between control unit and process peripherals have to be part of the system description.

Based on past investigations [1] and on experience gained in the area of aerospace developments and with the aim in mind to implement an as much as possible low cost approach, a system has been defined and implemented the main features and potential applications of which are presented herein.

2. Data Handling System Requirements

2.1 General.

A data handling system concept should be

1. adaptable in a non-autonomous configuration to multi-purpose on-board computers
2. modular in design
3. ruggedized
4. reliable.

It should have for most applications

1. low power consumption
2. low weight and volume.

The functional requirements to be fulfilled are basically to acquire, store and process data and to control peripheral processes. For that purpose a freely programmable Data Handling System (DHS) is required. Furthermore, at least, a possibility must exist to reprogram the system during in-flight via telecommand link and to use it for PCM-data encoding in telemetry applications. Asynchronous tasks, which have to be performed on external requests in real time should be possible, so that the system must have a respective powerful interrupt capability. Direct memory access and extended arithmetic and logic function capability should also be system inherent. For reasons of modularity and transfer speed of data words, a parallel bus concept must be favoured. For cost reasons two technological approaches should exist, one for the application in space missions with high reliability requirements, one for application in other missions like in aircraft and tank data handling. These approaches should be functionally identical.

2.2 Detailed Functional Requirements. After extensive evaluations of a variety of different missions [2] the following functional requirements were established, valid for a respective data handling system.

Typical transfer rate on the data bus		500 k Bit/sec
Typical execution times:	for ADD/SUB	< 10 µs
	for MULT/DIV	<100 µs
	for logic functions	< 10 µs
Transfer rate on the bus		< 10 µs
Program jump into subroutines		< 5 µs
Storage capacity		> 1 K words
Power consumption		< 5 W
Weight		< 2 kg
Volume		$< 2 \cdot 10^3\ \text{cm}^3$

The spectrum of necessary functional modules was determined to be:

> central processor, data storage module, A/D resp. D/A converters, synchro-to-digital converters, multiplexers, real time clock, digital inputs and digital outputs (serial and parallel), counters, PCM data serializer, programmable amplifiers,

only to mention the most recurring ones.

3. System Implementation

3.1 General System Configuration. On the basis of the derived system requirements, the complete data handling system “MUDAS” has been implemented. “MUDAS” stands for Modular Universal Data Acquisition System. The system philosophy more generally spoken, has similarities to ‘CAMAC’ [3] which is a commercial standard for real time process peripherals and their interfacing to a computer.

In the basic MUDAS system configuration (Fig. 1), 32 peripheral modules can be controlled at maximum by a central control unit via parallel bidirectional dataway.

Any type of module function can be realized since only mechanical layout and the electrical interface to the common dataway is defined in the MUDAS standard. The central unit can be either a MUDAS Dataprocessor or any type of on-board computer. The MUDAS dataway is a centralized real time bus consisting of a 16 bit bidirectional data bus, a control bus and four power lines. The signal timing on the bus and the pin assignment are standardized. The instruction word has 9 bits, three of which determine the function of the module, one indicates the direction of transfer, and five are available to address the modules. The module address is decoded in the dataway - controller and passed to the modules on single address lines.

The complete bus is characterized by Table 1.

The modules execute the desired functions controlled via the MUDAS dataway. A module consists of a dataway coupler meeting the dataway specifications, a functional unit performing the required functions and a process coupler for adaptation to peripherals. Most of the modules are realized with one board of the MUDAS standard, several boards are provided for more complex modules.

There are three types of data transfers between the central processor and the modules. Read and write cycles are programmed I/O transfers. Different types of interrupt transfers are used to handle module alarms.

At the beginning of a programmed write cycle (see Fig.2), the dataway controller provides simultaneously the 16 bit data word, the address N for the respective module, the 3 function bits for different possible module functions, and the transfer direction bit on the bus. Read and write cycle are characterized by the transfer status bit either being ‘high’ resp. ‘low’. After one time unit (1 µs) the function strobe FS is raised to high level by the processor, at this time the data are valid. If the module is ready to accept the data, it transmits the acknowledgement signal FSA, where upon the dataway controller recalls FS and the module FSA respectively.

Fig. 2 shows the timing on the data bus, based on a 1 MHz clock frequency. Read cycle means that a data word is being transferred from a specific module to the central processor. In ‘write’ cycle data words are transferred in the opposite direction.

Each module having an alarm control logic can trigger an interrupt in the processor. The order of priority is fixed by a series arrangement of the modules generating an alarm. A module alarm is passed on to the processor regardless of the present status of the system. As soon as the alarm can be processed, an alarm acknowledgement signal ALA is issued as a global address.

3.2 The Data Processors. There are existing three different data processors of different complexity. The first one of this series is a simple one (on one board) and only suited for pure logical functions, data handling and data exchange. A second one is already able to treat arithmetic operations. A third type of processor (DP 430) is being described herin. This one has also extended arithmetic with hard wired multiply/ divide capability.

Hardware. The DP 430 consists of an arithmetic and logic unit a register block, interrupt logic, a DMA control device, data I/O interface and associated gating and timing circuitry (Fig. 3). Fig.4 shows the hardware arrangement of Fig.3.

The Arithmetic and Logic Unit (ALU), controlled by a separate control logic, performs full arithmetic, logical and shift operations. Three registers (A, E, MD) store the temporary and final results of an ALU operation. The register block consists of 16 registers which are program accessible. 15 registers are used as accumulators and register Ro contains the program counter.

The S-Register stores during an instruction cycle the instruction that is to be executed by the ALU and the run down control respectively.

There are four flags which can be cleared and set under program control. Three flags indicate internal events e.g. carry and overflow. An external condition line EC can either be used for interrupt or for an external extension of the flag register. The fourth flag can be set by external events only.

All necessary timing and control signals derived from an internal, adjustable oscillator are generated by the Control Logic Unit. External accessible inputs and outputs are provided for control and monitoring of the actual state of the Data Processor. The interrupt logic controls the interrupts which can be direct or indirect, initiated by an instruction or by external conditions.

The data processor has a simple direct memory access capability for block data transfer from the memory to a module and vice versa. The DMA request and the transfer direction must be defined by the module.

The dataway coupler handles data, I/O instructions and control signals for data transfer with peripheral devices. It consists of a 16 bit bidirectional data bus, an address decoder which provides 32 single address lines for peripheral units, a three bit parallel functional bus and four transfer control and timing signals.

The memory is connected to the main bus via a standard memory interface. The 16 bit information is handled as data or as memory address, that means a maximum of 64 k words are addressable. Due to the asynchronous timing, any memory type, independent of access time, can be interfaced with the Data Processor.

Essential Characteristics of the DP 430

• Type	General purpose, program controlled
• Logic	CMOs
• Operation	Binary parallel
• Word length	16 bits
• Hardware registers	16 x 16 bit RAM
• Arithmetic	Two’s complement, fixed point
• Instruction set	30 basic instructions

	logic and full arithmetic facility
• Execution times	Branch 5 - 7µs
(at 1 MHz)	Add/Subtract 7 µs
	I/O transfer 9 µs
	Multiply 27 µs
	Divide 49 µs
• Clock	Adjustable up to 1 MHz
• Program interrupt	3 interrupts
• Input/Output	1 module bus (16 bits,parallel, tristate)
	1 memory bus (16 bits, parallel, tristate)
• Word rate	Module transfer 110 kHzmax.
	Direct Memory Access 250 kHz max.
• Memory	Semiconductor or Core Memory up to
	64 k addressable
• Dimension	4 pluggable PCB's with 122 x 105 mm^2
• Weight	240 g (memory excluded)
• Power consumption	less than 1 W at 1 MHz
• Power supply	10 V
• Temperature range	-40°C to +100°C (operating).

Instruction Set. The basic instruction set (Table 2) of the processor can be classified into

transfer instructions (to and from the modules)
set instructions
branch instructions
register handling and execute instructions*
load and store instructions
substitute instructions

* The register execute instructions EFA, EFB are corresponding with the ALU (Arithmetic and Logic Unit) and initialize 32 different arithmetic operations (for example add/subtr.), whereas EFLA and EFLB initialize resp. logical operations in the ALU (see Table 2).

Software-Aids for DP 430. A software development system with which data processor programs can be set up and logically tested on host computers is available for the MUDAS Data Processor. The software development system consists of an assembler, an output program, a simulator, a debugger and a dump program. While the assembler and the output program are essential in order to set up a data processor program, the simulator, debugger and dump program are to be regarded as test aids that are normally part of the monitoring software of process control computers.

The assembler converts the data processor program as set up by a user in a mnemonic code, into the data processor code and files this code in an auxiliary memory. The auxiliary memory can be either resident in the computer system's core memory or simulated on a magnetic disc.

The Output Program codes bit by bit the data processor instruction words in the ASCII code with a parity bit, onto a paper tape or a magnetic tape. The paper tape or magnetic tape then acts as a data carrier for the PROM programming maschine. The data format is determined by the type of PROM programming facility and the PROM used.

The simulator is to simulate the data processor itself and executes the data processor program filed by the assembler in the auxiliary memory. Data transfers and the clearing or setting of flags can be simulated, as well as alarms and interrupts. The simulator takes the instruction under the program address in the auxiliary memory and executes it in the same way as the data processor would do. After execution of the instruction, the register contents and the flag status are printed on the high-speed printer. If halts have been set at certain program addresses with the aid of the correction program, before beginning the simulation, the run-down of the simulation will be interrupted at each halt address. The control instruction interpreter (housekeeping program) can now be used either to set new halts or clear old ones , to modify the contents of registers or program memories or merely to dump the memory and register contents. Dump programs for memory and register read-out during software development and debug-programs to change the register contents of the processor are available.

Apart from these already developed programs the development of a mathematical program library and of a real time operating system is under way.

3.3 The Peripheral Modules. Each module which meets the interface specification of the bus can be plugged into a mother board by means of a MIL-strip connector or if this connector is not tolerated in special applications this connector can be replaced by soldered wire or socket connectors. The mother board contains the data bus and all necessary interconnections between modules and processor. The number of standard modules developed are for example:

A/D converters
D/A converters
Synchro to digital converters
Multiplexer
Parallel/serial input/output modules
Counters
Real time clock

Telemetry encoder
Amplifier modules

Special-to-type modules can easily be incorporated in the system if the interface to the data bus and the proper timing is met by the designer.

3.4 Technology and Mechanical Dimensions. The MUDAS modules are built-up of one or more daughter boards with the dimensions 122 x 105 mm^2. The preferred interconnection between daughter boards and the mother boards is performed by MIL-strip connectors with 81 pins.

The thickness of PCB’s is 1 mm. Weight of one board is approx. 60 gr.

Especially for power reasons C-MOS logic is used wherever possible. Therefore the signal levels on the dataway are specified to 10 Volts.

The modules can be equipped with parts that are obtainable in qualified form according to specifications noted in PPL 12 or MIL-Stand.883 Class A,B or C and JANTX and MIL-ER. Since the boards are also obtainable in various quality classes, various different quality requirements can be met for the modules and thus for the overall system.

Environmental conditions:

Operating temperature range	-40°C to +100°C
Storage Temperature range	-65°C to +125°C
Thermal Vacuum	10^{-5} Torr
	(-40°C to +80°C)
Vibration:	
sinus	30 g (25 Hz to 200 Hz)
random	17 G RMS (overall)

4. Redundant System Configuration. Reliability requirements for the overall system may necessitate redundant configurations. Due to the modular design the system reliability of MUDAS can be increased by both module and system redundancy. Fig.5 shows the general block diagram of a redundant MUDAS configuration. Via dataway resp. M-Bus-selector each of the two processors can control the two buses and can interact with two memories. A reconfiguration of the system can be achieved either by external command or in case of module failure under program control. Other approaches which regard only partial redundancy can be implemented as well.

5. Serial Bus Concept. The parallel MUDAS bus described in chapter 3 is limited to 1 m length and is thus well suited for concentrated systems.

In case of distributed system configurations which have to perform data exchange between each other, a serial transfer procedure is being considered as shown in Fig-6 . This serial transfer procedure is based on specifications laid down in MIL STD 1553 A. The transfer medium is a twisted pair resp. coax. cable. The transfer of information is on half duplex procedure. Max. bitrate is considered to be 1 Mbit/sec. (Manchester Bi-Phase). Each data word to be transferred (16 Bit) contains 3 synchronization and 1 parity bit. Command-status and data-words can be transferred and are distinguished by the synchronization bit pattern. Interrupt handling via bus is also possible. DC-isolation on the bus is attained by pulse transformers. In case of failure in one receiving station, a short circuit protection is provided in the transformer part. The interconnection of intelligent and nonintelligent parts of the system can herewith be performed as well.

6. Application of the MUDAS System.

6.1 General. The MUDAS Data Handling System can be applied in a large field. This is due to the fact that several different technological and mechanical alternatives are foreseen and several processing units do exist.

In recent projects the MUDAS System is used for example as telemetry encoder in sounding rockets, balloon gondolas and tanks; as flight data recording system in military aircrafts. Furthermore applications in data handling systems of data collection platforms for meteorological satellites and combined data acquisition and control for communication satellite projects. Further applications are envisaged in the spacelab program for the control loop of the Instrument Pointing System.

From the existing spectrum of applications one example shall be presented in more detail below.

6.2 Example for a typical Application. A typical application for the MUDAS Data Handling System is described herein. One MUDAS system is configured for three parallel tasks, one of which is data acquisition in a sounding rocket payload (‘Skylark’ type), the second is telemetry encoding and the third is attitude control and stabilization of the payload itself. Whereas in conventional approaches this tasks would have to be performed by separate subsystems, the flexible modular concept of MUDAS allows a combination of all tasks.

Fig. 7 shows the block diagram of this data handling and -processing system.

Analog signals are handled by three multiplexers of 32 channels each; the output signals of the multiplexers are converted by a 12 bit ADC. The digital housekeeping data to be acquired are processed via digital input module of 6 x 8 bit.

The control of the gas nozzles is been achieved via digital output. Thus 8 control nozzles can be activated causing control moments around the lateral axis x, y resp. around the spin axis z of the sounding rocket payload. If pregiven thresholds of the attitude sensors (rate gyros) are exceeded after a spin decay has been performed by a yo-yo system, the MUDAS system activates the control nozzles of the cold gas system according to a pregiven control law.

The transfer of a serial Split Phase coded data-stream is achieved by the PCM output module. The acquisition of attitude control data for telemetry encoding and control processing must be performed in this combined system only once. Data are transferred under processor control to the PCM output module and simultaneously data are processed to derive the control signals for the nozzles to achieve attitude stabilization. The telemetry frame is configured to transfer data by a frame frequency of 20/sec.

Subframe frequency is established to be 5/sec. Format length is 172 words. The main frame consists of 43 words. The number of bits per word is 12 + 1 parity. Synchronization word length is 35 bits. Bit frequency is 11,180 kbits/s.

As no reprogrammability from ground is requested for that mission, all programs are stored in EPPROM’s (eraseable, programmable read only memories). Apart from an interface module to switch the nozzle on and off, the whole system relies on standard MUDAS modules.

The Program. According to the given tasks the following program modules-have been established:

a) **Initializing Phase**

After initializing of the system, ‘Clear’ and ‘Reset’ instructions are given to all peripheral modules to create defined starting conditions. Furthermore the registers of the processing are set into defined conditions.

b) **Data Acquisition**

The data acquisition either runs via a/d-converters with multiplexer inputs resp. via digital inputs with byte oriented registers The defined timing for the sampling of all signals and the resp. conversion has to be clearly met.

c) **Data-Processing**

All operations correlated with the arithmetic and logic unit have to be performed by this program, i.e. mean value and difference value computing generation of the sync. code and the identity words in the subframes.

d) **Attitude Control**

By comparing the sensor data with pregiven threshold values the attitude control law has to be derived and the control nozzles be activated.

e) **Data-Encoding**

The data output via PCM-module is activated by this program.

f) **Test Program**

This program facilitates the check-out of the system during integration. It applies for example test signals to the input of the multiplexers and compares the processed data with pregiven values. Thus a fast automatic check-out is possible. The output of signals is performed via PCM output module to the telemetry receiving station. Errors in the system can thus be detected on module basis by no measuring effort.

This test program is presently foreseen only for integration purposes, but could easily be extended to onboard testing as it is system inherently stored in the memory.

The time determining factor in the program is the PCM output module which sends word requests in a fixed time interval to the data processor by interrupt procedure. Thus a continious serial data-stream to the TM-transmitter can be achieved.

The whole program for the described task requires a 1 k/16 bit memory capacity for approximately 1000 instruction words. It is written in assembly language simulated and tested under the use of existing and described software aids of the system.

Conclusions. The use of programmable modular data handling systems facilitates the integration and set into operation of complex systems. The reduced development risk is mainly due to the fact that clear interfaces between the process peripherals, the processor and the outer world are established. If clear software tools and their application are available as in the described system, an easy and direct way for implementing a great variety of different tasks can be achieved.

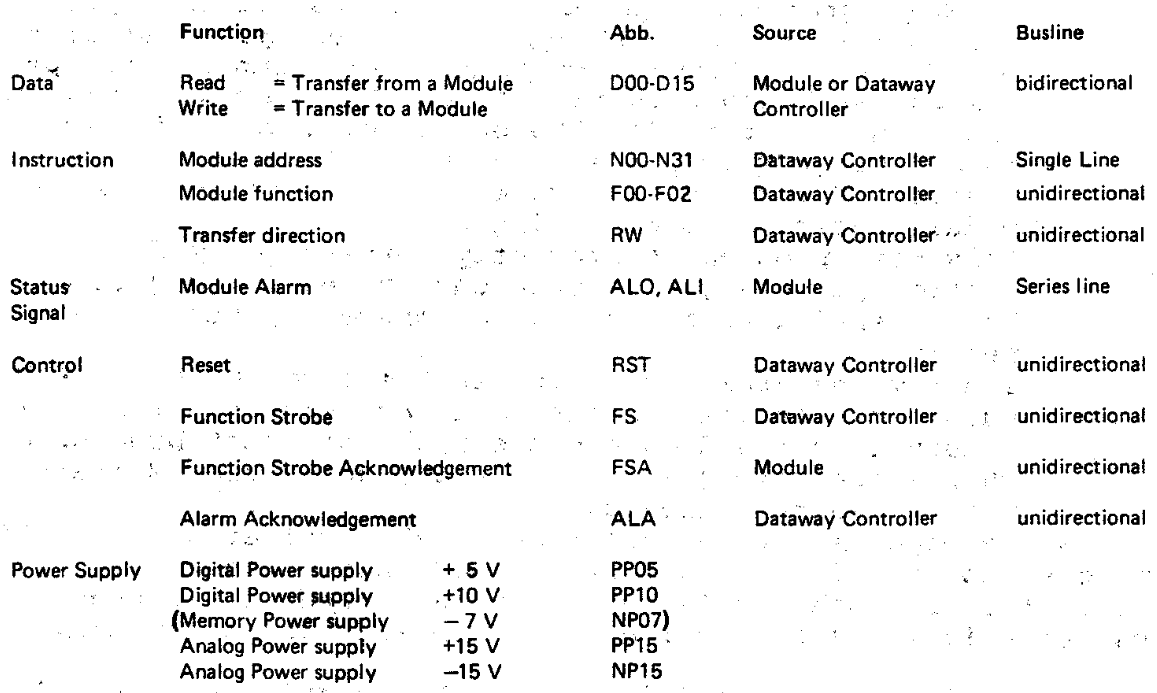
References

[1] Fröh1ich H.; Schweizer G., “Contribution to the Design of Future On-Board, Dataprocessing Systems for Scientific Spacecraft Experiments” Proceedings of the International Telemetering Conference 1972, Instrument Society of America, Pittsburgh PA, USA

[2] “Anfotderungsanalyse für modulates, Datenerfassunassystem in Raurfahrtanwendungen” Study report presented under Contract Nr. RV11-V84/73-TL 20, DFVLR-BPT by Dornier System GmbH, FRG, Nov. 1975

[3] Costrell, L. “CAMAC Instrumentation System Introduction and General Description” IEEE Transactions on Nuclear Science, Vol 18, No. 2 April 1971.

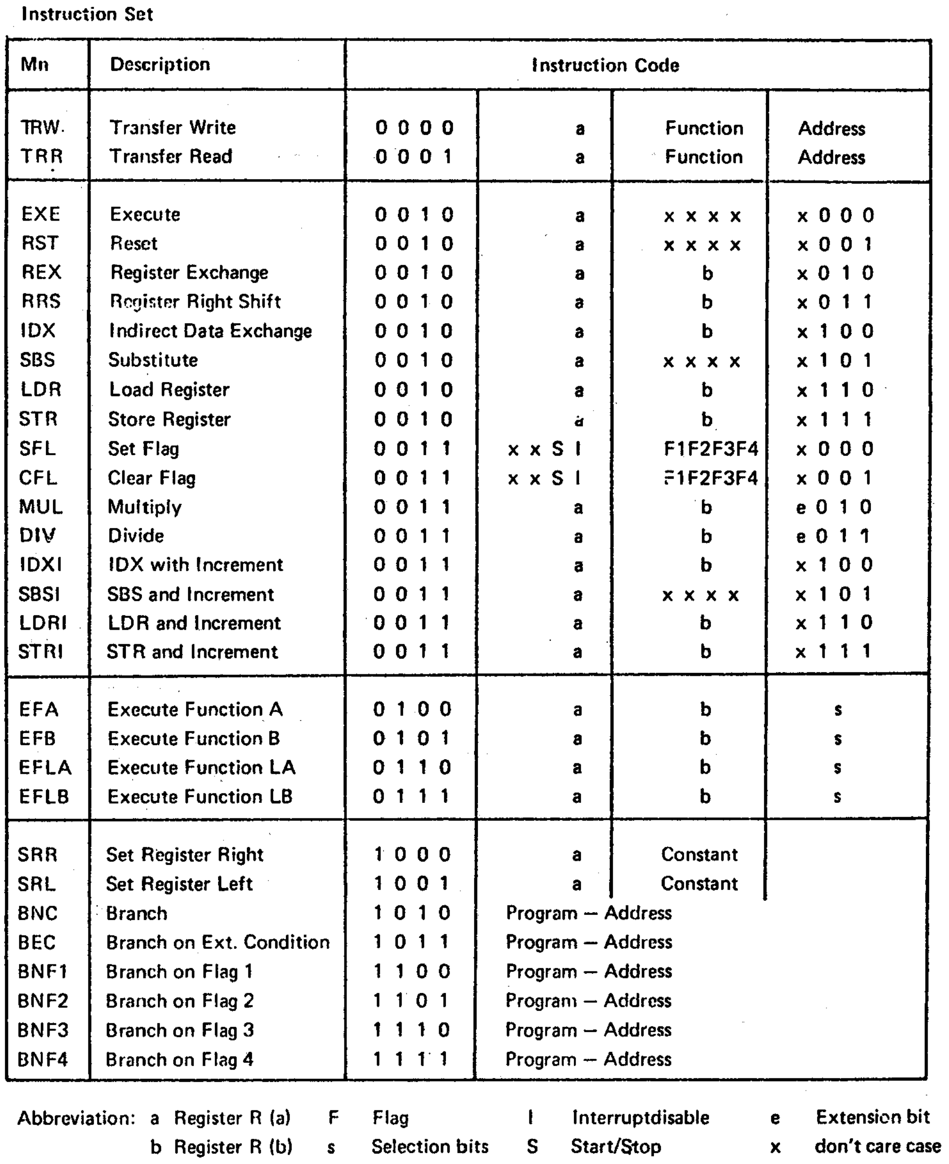
TABLE 1



	Function		Abb.	Source	Busline
Data	Read = Transfer from a Module Write = Transfer to a Module		D00-D15	Module or Dataway Controller	bidirectional
Instruction	Module address		N00-N31	Dataway Controller	Single Line
	Module function		F00-F02	Dataway Controller	unidirectional
	Transfer direction		RW	Dataway Controller	unidirectional
Status Signal	Module Alarm		ALO, ALI	Module	Series line
Control	Reset		RST	Dataway Controller	unidirectional
	Function Strobe		FS	Dataway Controller	unidirectional
	Function Strobe Acknowledgement		FSA	Module	unidirectional
	Alarm Acknowledgement		ALA	Dataway Controller	unidirectional
Power Supply	Digital Power supply	+ 5 V	PP05		
	Digital Power supply	+10 V	PP10		
	(Memory Power supply	− 7 V	NP07)		
	Analog Power supply	+15 V	PP15		
	Analog Power supply	−15 V	NP15		

MUDAS Dataway Signals

TABLE 2



Instruction Set

Mn	Description	Instruction Code			
TRW	Transfer Write	0 0 0 0	a	Function	Address
TRR	Transfer Read	0 0 0 1	a	Function	Address
EXE	Execute	0 0 1 0	a	x x x x	x 0 0 0
RST	Reset	0 0 1 0	a	x x x x	x 0 0 1
REX	Register Exchange	0 0 1 0	a	b	x 0 1 0
RRS	Register Right Shift	0 0 1 0	a	b	x 0 1 1
IDX	Indirect Data Exchange	0 0 1 0	a	b	x 1 0 0
SBS	Substitute	0 0 1 0	a	x x x x	x 1 0 1
LDR	Load Register	0 0 1 0	a	b	x 1 1 0
STR	Store Register	0 0 1 0	a	b	x 1 1 1
SFL	Set Flag	0 0 1 1	x x S I	F1F2F3F4	x 0 0 0
CFL	Clear Flag	0 0 1 1	x x S I	F1F2F3F4	x 0 0 1
MUL	Multiply	0 0 1 1	a	b	e 0 1 0
DIV	Divide	0 0 1 1	a	b	e 0 1 1
IDXI	IDX with Increment	0 0 1 1	a	b	x 1 0 0
SBSI	SBS and Increment	0 0 1 1	a	x x x x	x 1 0 1
LDRI	LDR and Increment	0 0 1 1	a	b	x 1 1 0
STRI	STR and Increment	0 0 1 1	a	b	x 1 1 1
EFA	Execute Function A	0 1 0 0	a	b	s
EFB	Execute Function B	0 1 0 1	a	b	s
EFLA	Execute Function LA	0 1 1 0	a	b	s
EFLB	Execute Function LB	0 1 1 1	a	b	s
SRR	Set Register Right	1 0 0 0	a	Constant	
SRL	Set Register Left	1 0 0 1	a	Constant	
BNC	Branch	1 0 1 0	Program – Address		
BEC	Branch on Ext. Condition	1 0 1 1	Program – Address		
BNF1	Branch on Flag 1	1 1 0 0	Program – Address		
BNF2	Branch on Flag 2	1 1 0 1	Program – Address		
BNF3	Branch on Flag 3	1 1 1 0	Program – Address		
BNF4	Branch on Flag 4	1 1 1 1	Program – Address		

Abbreviation: a	Register R (a)	F	Flag	I	Interruptdisable	e	Extension bit
b	Register R (b)	s	Selection bits	S	Start/Stop	x	don't care case

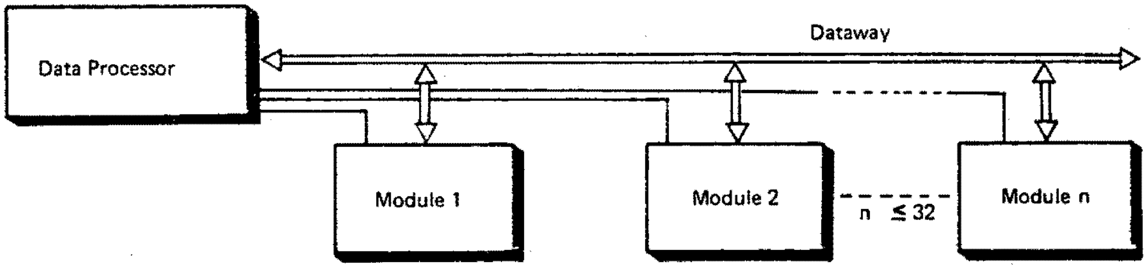


Figure 1 'MUDAS' System Configuration

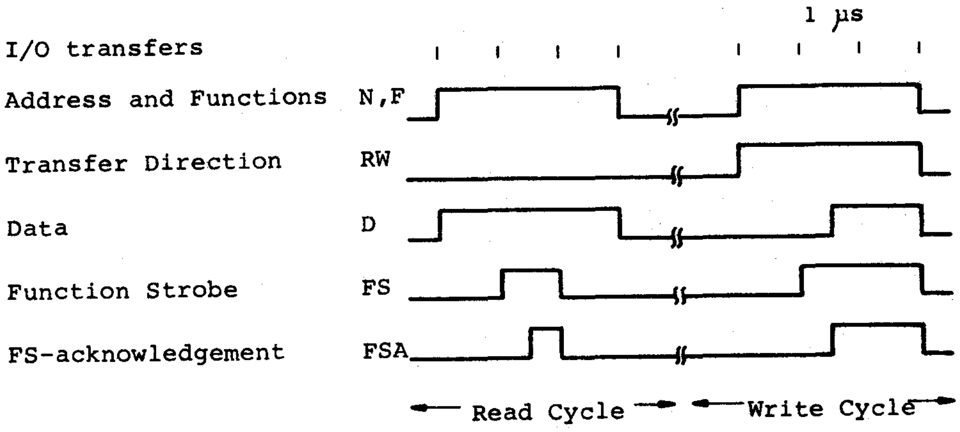


Figure 2 Timing On the Data Bus

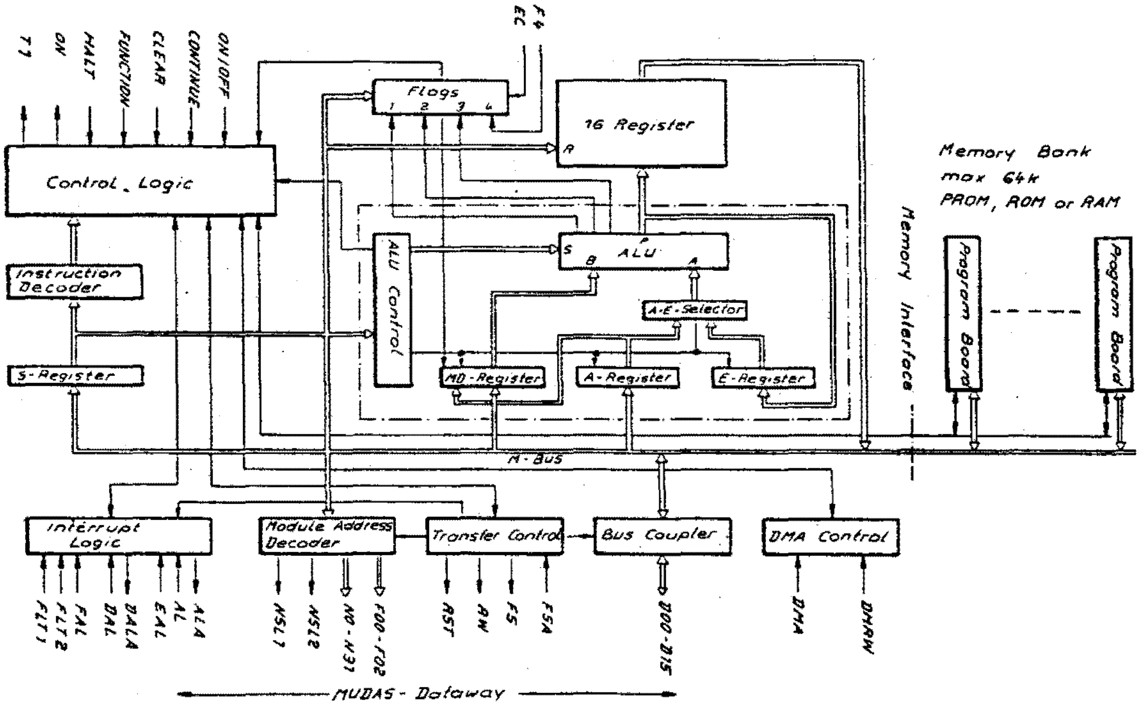


Figure 3 Block Diagram of MUDAS Data Processor (DP 430)

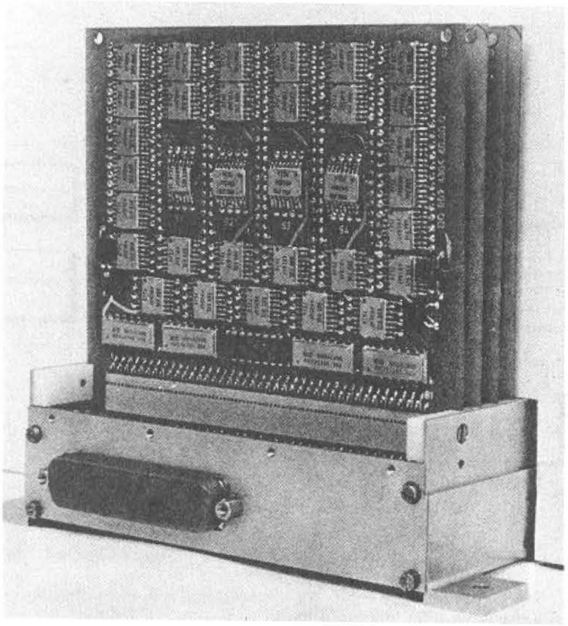


Figure 4 MUDAS Data Processor DP 430 Arrangement of the Boards

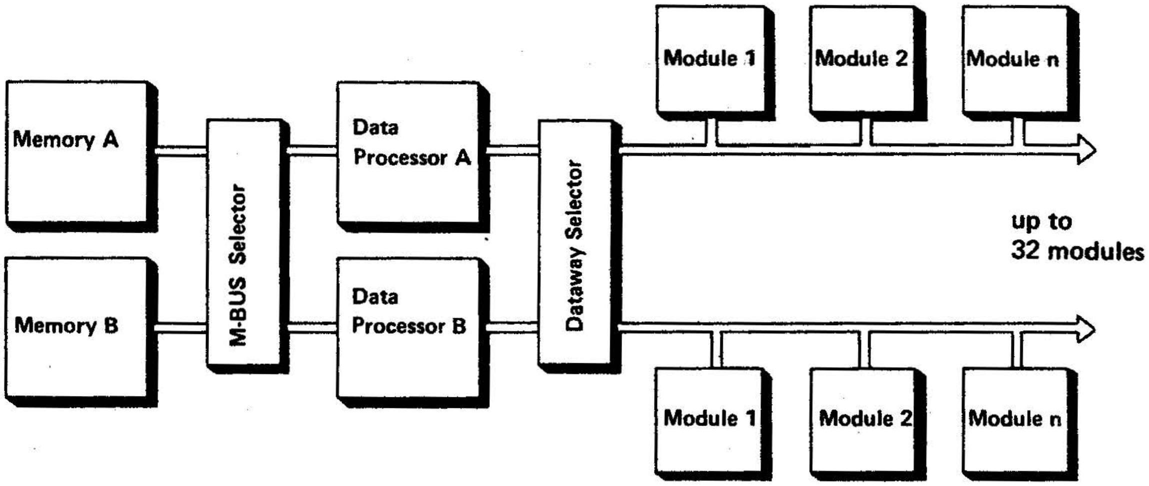


Figure 5 Redundant MUDAS Configuration

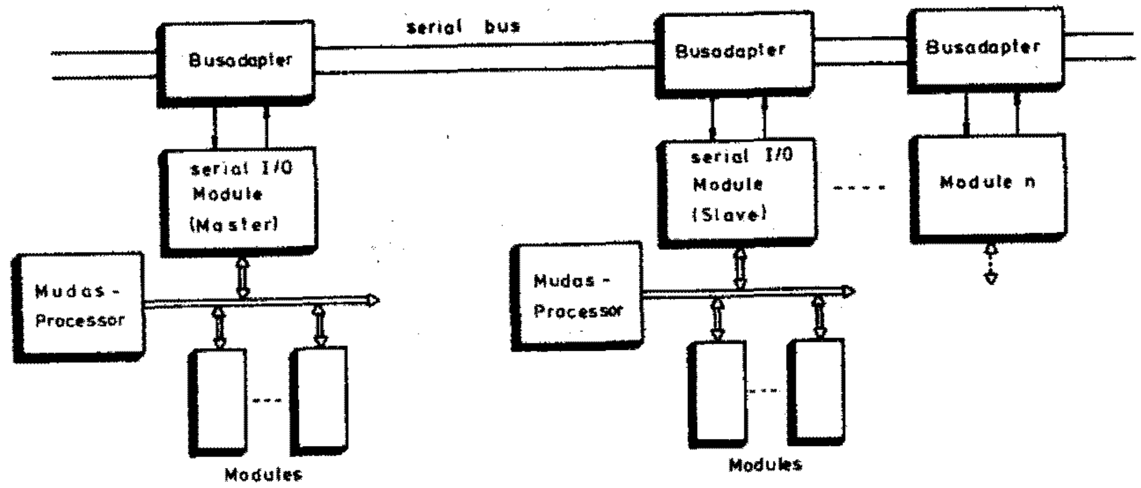


Figure 6 Typical Arrangement of 'Mudas' systems interconnected by a serial bus

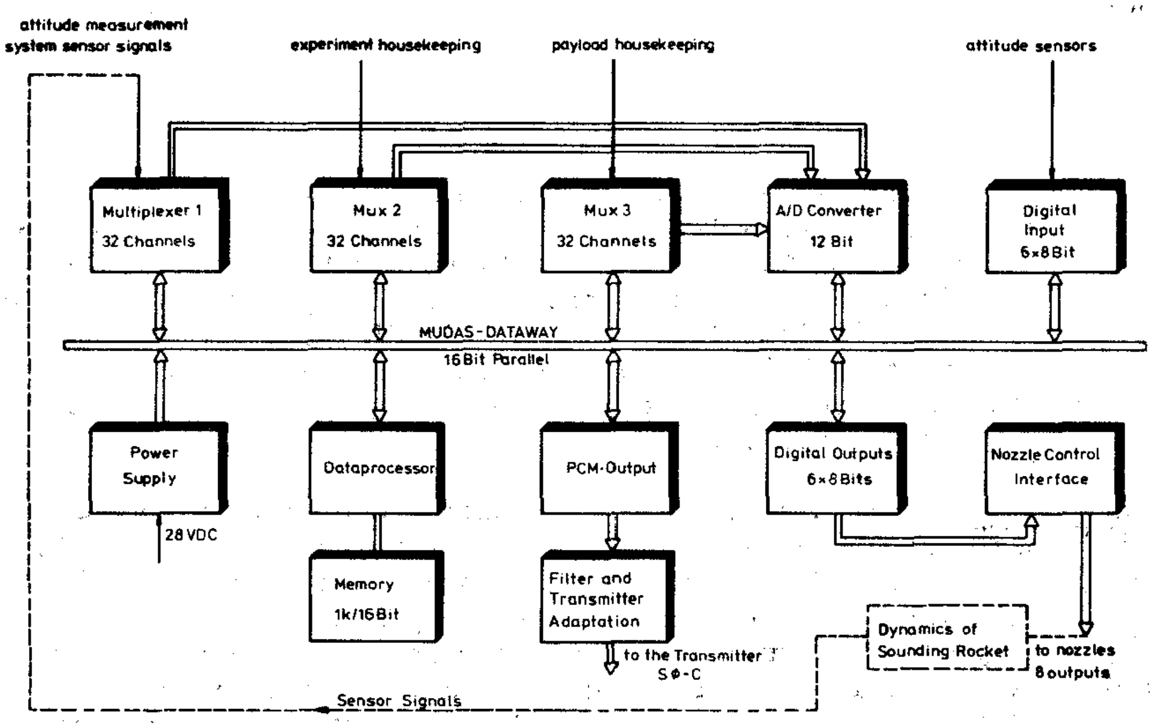


Figure 7 Combined Telemetry Encoding and Payload Stabilization System