

Design of a High Reliability Self Diagnosing Computer Using Bit Slice Microprocessors

S. Sanyal and P.V.S. Rao

Computer Systems & Communications Group, Tata Institute of Fundamental Research, Bombay-400005, India

This paper describes a processor built to meet the requirements for a highly reliable and ruggedised digital computer. Innovative techniques were used to achieve high performance without using very high reliability components or redundancy at the circuit level. The processor therefore was designed using moderately reliable components (mil B standard) with microprogrammed control logic and powerful built-in microdiagnostic capabilities. It was successfully used for two major applications: a digital switching system and a data acquisition and processing system for a weather radar.

Keywords: Reliability, Microprogramming, Microdiagnostics.

1. Introduction

A need arose for the development of a rugged digital computer to form the nucleus of complex online systems required for two major applications: a general purpose medium-size digital switching system [1] and a data acquisition and processing system for a weather radar.

These applications called for high computational speed (250 k instructions/second), reliability ('MEAN TIME BETWEEN FAILURE' greater than 1000 hours), ruggedness, ease of maintenance ('MEAN TIME TO REPAIR' less than 30 minutes) and easy transportability. The machine which was developed to meet these requirements, though tightly constrained in terms of performance, provided a stimulating challenge and ample scope for design innovation.

The basic objective of developing a highly reliable computer was achieved, not through the usage of highly reliable components, but through a novel and innovative microprogrammed scheme. In this design, microdiagnostic facilities were provided for powerful fault-detection and fault-location capabilities.

2. Design Options

2.1 Functional Design

To achieve the desirable level of reliability and speed, several design options were considered:

(a) In a standard design using very high reliability components (MIL A Standard), the cost factor is prohibitive. Also a standard design using MIL A standard components still presupposes continuous functioning of the components. The probability of failure is reduced, but the time required for repair does not decrease.

(b) Using moderately reliable components (MIL B Standard) in a design which provides for automatic fault identification permits quick replacement of the faulty submodule. A high Mean Time Between Failures and a low Mean Time to Repair can both be achieved.

(c) Triple Modular Redundancy (TMR) (where processors and checking circuits are triplicated and a majority 2 out of 3 voting logic is used to detect a faulty processor) would be extremely costly. It can also turn out to be a self defeating exercise as the rate of failure increases due to the additional hardware.

2.2 Choice of Technology

Three options were considered regarding the choice of architecture and technology:

(a) Standard fixed format processor: The functional blocks for the Data Path and Control Section shown in Fig. 1 are available in single chips in several standard microprocessor families: INTEL 8085, ZILOG Z80 or MOTOROLA 6800.

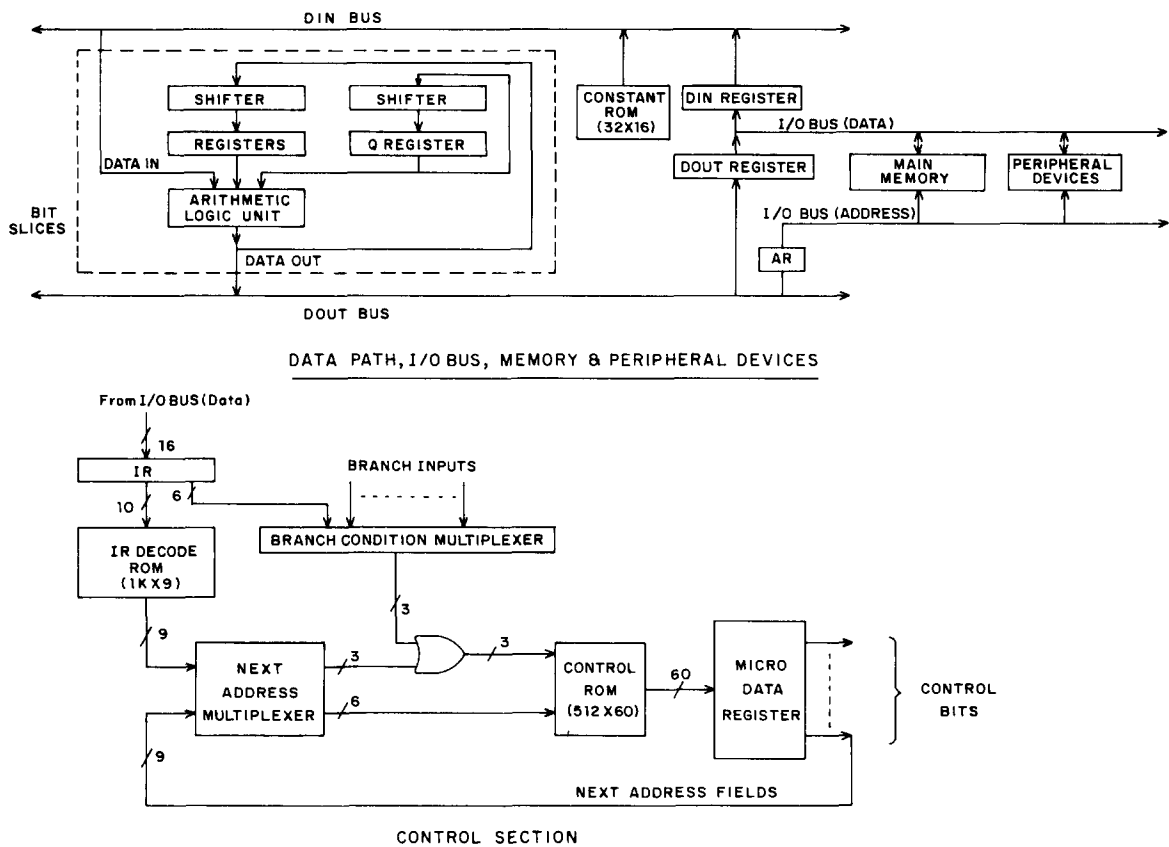


Fig. 1. BLOCK DIAGRAM OF THE 16 BIT COMPUTER

The Standard Processor approach has the following problems:

- i. The Fixed Format Instruction Set and Architecture of these microprocessors do not allow the designer any flexibility for implementing his own architecture.
- ii. The processing speed of these microprocessors (which typically have an 8 MHz clock rate) cannot cope with the processing load generated in a real time control application.
- iii. Additional reliability features cannot be built into such a design because the control section is 'hard wired'.

(b) MSI and SSI implementation: It is possible to design processors using Medium and Small Scale

Integrated circuit (MSI & SSI) logic modules (Registers, Counters, Gates etc.). This approach provides adequate scope for original design of the processors and for incorporating the desired reliability features. It also makes it possible to achieve higher processing speed by using fast components: Schottky Transistor Transistor Logic (S-TTL), Emitter Coupled Logic (ECL) and so on.

(c) Design based on bit slice microprocessors: Circuits consisting of hundreds of MSI and SSI chips can be replaced by a few Bipolar Bit Slice Microprocessor Chips and appropriate support chips. The 16 bit data path in Fig. 1 can be completely designed using four 4 bit slices. This approach yields a speed improvement by a factor of 10 over fixed format microprocessors.

Another advantage is modularity. Bit-slice mi-

croprocessors are normally four-bit wide. Computers of any wordlength can be configured using the required number of slices.

Normally, bit slice microprocessors are designed with Schottky and Lowpower Schottky TTL's. Hence, it is possible to achieve microcycle times down to 100 ns from these chips. They are also useful for making specialized controllers (e.g. disk controllers).

A comparative study of different bit-slice processors may be found in Table 1. It is obvious from Table 1 that Am 2901 is the most suitable choice.

2.3 Diagnostic Methodology

Self checking of the system is carried out on a time shared basis with the actual processing work. Whenever the operating system goes to an 'idle' state, waiting for a process to be initiated, the diagnostic routine is invoked. This is assigned a low priority so that real time processes of higher priority do not suffer due to unavailability of the system.

In the diagnostic phase, microdiagnostic routines check the data path and the control logic completely. These are much faster than conventional diagnostic routines written in machine language. Another advantage of microdiagnostic routines is that they can check hardware very effectively at a much finer level. In addition, hardware checkers

were incorporated selectively to provide a direct indication of some types of failures.

3. Implementation

3.1 Description of the Basic Central Processor Logic

Fig. 1 shows a block diagram of the basic processor (datapath and control section). Am 2901 (Advanced Micro Devices) bit slices are the main components of the datapath. Each 4 bit slice contains sixteen four bit two port addressable registers, a four bit Arithmetic Logic Unit and shifter logic (capable of 1 bit right or left shift or no shift); a four bit expandable Q register provides the facility for double word operations.

3.1.1 Arithmetic Logic Unit

The arithmetic logic unit provides appropriate flags for storage of control information resulting from an operation:

(a) the sign (N) bit is set if any arithmetic operation generates a 2's complement negative number.

(b) the zero (Z) bit is set if the result of any operation is zero.

(c) the carry (C) bit is set if the result produces a carry.

(d) the overflow (V) bit is set if the result of any operation exceeds the permitted range [2's complement numbers can range over $+(2^n-1)$ to -2^n if the wordlength is n bits].

3.1.2 Input/Output Bus

The Input/Output (I/O) bus provides the means for the processor to communicate with the peripheral devices. Internally, the DIIn Bus and the DOut Bus are used for addressing and data transactions. The DIIn Register, DOut Register and the Address Register (AR) function as the respective buffers.

3.1.3 Control Section

(a) *The control memory*: This Read Only Memory (ROM) holds the main microprogram routine for

Table 1.

Manufacturer	Device	Logic family	Bit slice width	Speed	Power supply levels required	PCB design	Availability of Mil SPEC version
Advanced microdevices	Am 2901 (AMD hand-book) 1976	Low power Schottky TTL	4	100ns	1	Not difficult	Available
Signetics	3000	Shottky TTL	2	100ns	1	Not difficult	Available
Motorola	MC10800	ECL	4	55ns	2	Difficult	Not available at time of design

instruction execution, 'exception' handling, console switch control and microdiagnostics. Instruction Register Decode (IRD) circuitry decodes the instructions and generates the appropriate starting microaddresses in each case.

(b) *The basic design:* The design incorporates a number of special features, some of which are explained here:

- i. *The Pipeline Microdata Register:* This register is situated at the output of the ROMs; it clocks and holds microbits for one clock cycle. During this time, the contents of the next microaddress are also acquired and kept in a pipeline at the input of this register. These are suitably modified in case of micro branching.
- ii. *Next Address Branch Condition Synchronization:* It is important to ensure that any 'microaddress modifying condition' is established within a reasonable time before the advent of the rising edge of the next clock. The branch condition could be an independent signal (asynchronous with respect to the processor clock; e.g. a 'RESET' monoshot getting de-asserted). Any such external condition is always synchronized with the microprocessor clock which strobes the microdata register. This ensures that the next address field of the ROM does not change at any arbitrary time.
- iii. *Variable Clock Period:* In a microprogrammed computer, one microword is normally gated into the microdata register during each clock cycle. It is essential that the input signal levels for every microword should stabilise before clocking. This in turn requires that the signals defining the microaddress and the conditions for modifying the microaddress should stabilise within a specified time after the start of a clock cycle. For different microcycles, the microaddress branching conditions take different times to settle. Consequently, four different clock cycles were provided for: 150 ns, 200 ns, 275 ns and 350 ns; these were controlled by two microbits.

iv. *Variable Bus Signal Deskew:* In the asynchronous I/O bus, it is necessary to allow for some delays between control signals and the corresponding qualifying signals arising from skew along the long bus. It is necessary to deskew these signals suitably. A variable deskew facility was implemented for greater time saving. Since the memory sits in close proximity to the processor, the deskew time for memory was fixed at a low value (25 ns). The deskew time for peripheral devices was fixed at 150 ns since these are situated at a distance on the bus.

v. *'ABORT' Condition:* In case of unusual conditions such as illegal instructions, the normal microprogram sequence has to be aborted. This may be achieved by constant monitoring of all such conditions. If any of them is present, special hardware forces the 'present microaddress' to assume a predetermined value which points to suitable 'ABORT' microroutines.

vi. *Maintenance Facilities:* A synchronization signal for the Logic Analyzer is provided by comparison of a microaddress (tapped at the ROM address input) with a value that can be set from the switch register on the console. This enables generation of a visual history of different signals against time over a period of time (maximally 1024 X sampling clock period).

Single microstepping is possible; here, the clock stops after execution of each microstate. A SINGLE STEP switch on the console activates each transition.

Entry into any predetermined microroutine is possible; a known microaddress can be forced in from the Console Switch Register.

(c) *Fault Diagnosis:* A hardware part of the system [2,3] is designed with adequate redundancy to make it reliable. The microdiagnostic routine first checks this hardware. If there is a fault here, an indication is provided. If this is fault free, further testing of the rest of the system is done using this working 'hard-core'. The following additional checks are provided;

- i. *ROM bit failure:* The control memory has byte

parity (1 parity bit is used for every 8 bits). Presently available ROMs are 8 bits wide (the Fairchild 93448 bipolar Programmable ROM has 512 eight bit words). To enhance the reliability further, the fields of the microcode are interleaved with parity bits and data bits residing in separate ROMs. This lessens the probability of multibit failures due to defects on the chip.

ii. *Rom address checking*: A microaddress parity is generated to check for proper functioning of the ROM address decoders. The addressed microword stores the expected parity bit and faults can be detected on comparison.

ii. *Entry into and exit from the microdiagnostic phase*: The frequency with which the microdiagnostic routines are called is decided depending on the strategy adopted for diagnostics.

Entry into the diagnostic mode is triggered by setting two bits i.e. 'START MDIAG BIT 1' and 'START MDIAG BIT 2' sequentially, by means of two separate instructions. Chances of accidental entry into the diagnostic mode are thus minimised as it is very unlikely that both these special instructions would accidentally occur, one immediately after the other. These two special bits are cleared when control exits from the microdiagnostic routine. To ensure reliable operation, MIL A standard chips have been used for this subunit.

iv. *Arithmetic Logic Unit (ALU) and General Purpose Register (GPR) checking*: This is done by a comparison check, using a check register. Known operands are fed to the Arithmetic Logic Unit. Various operations (arithmetic and logical) are executed and the results checked against expected values. Checking of the General Purpose Registers and the Q register of the bit slices is accomplished by loading known data into them and reading back. The shifting logic is also checked using known data.

v. *Special Purpose Register (Processor Status Word) Checking*: This register is extremely important and is therefore checked in two differ-

ent ways. Firstly, a known data pattern is written in and read back to check for possible faults. Secondly, the condition codes are set for various operations on known operands. Deviations from expected results would trigger an 'ERROR' condition.

vi. *Bus Controller Check*: The bus controller logic (not shown in Fig. 1) controls transactions between the memory and the peripheral devices. This logic is checked under microdiagnostic control by means of a 'dummy device': the bus device simulator. This checks all the bus signals and transactions.

In case of any fault being detected in this phase, the microdiagnostic routine branches to a 'Fault Detected' microprogram. This routine indicates to the external environment the presence of a fault and then halts the machine. In case of no fault, the machine goes back to normal operation.

(d) *Fault Location*: The fault location process starts with the checking of the 'hardcore'.

i. *Fault Address Resolution*: Fault resolution needs to be adequate to localise it to the smallest replaceable submodule, i.e. a card.

ii. *Hardware checking*: A ring counter is clocked continuously to provide the microaddresses in sequence to check the ROM addressing; faults in this region will be detected and located by either the byte parity checker or the microaddress parity checker. A suitable fault indication is generated in either case.

The next address logic is checked by generating different microbranch conditions; any unexpected 'ERROR' condition generates a fault indication signal.

The 'hardcore' checking circuitry directly indicates a failed circuit. The fault locating tests resolve the fault upto the card level, and set the appropriate indicators. This facilitates quick replacement, thus ensuring very low Mean Time to Repair.

4. Design Validation and Improvements

The basis design of the Central Processing Unit (CPU) was thoroughly pre-checked before implementation. The following improvements were effected as a fallout from this exercise.

(a) The Bus Controller of the CPU was initially designed using a microprogram control. The paper design indicated that the time taken for effecting a bus transaction would be much more than it would be for a conventional design. Hence, this portion was redesigned using hard-wired logic.

(b) The Input/Output bus was asynchronous. Any noise spike on the bus could in principle cause a malfunction. To circumvent this problem, the bus controller circuits were designed to be internally synchronous. Any signal on the bus was connected to the 'D' input of a Flip-flop and then clocked in with a high frequency (10 MHz) clock. This reduced the probability of a noise spike accidentally triggering the bus controller logic into action.

5. Self-Diagnostic Capability

The machine was fabricated and tested thoroughly. The diagnostic capability was subjected to rigorous testing. The following are some of the faults that were created intentionally for this test:

(a) The outputs of the ROM were separated from the input of the microdata register one at a time and temporarily grounded.

(b) The processor status word flags (N, Z, V, C,) were isolated and individually tied to high or low.

(c) The bit slice microprocessor output bits were isolated and stuck-at-1 or stuck-at-0 faults were forced.

(d) Faults were simulated on the I/O bus data and control lines.

(e) The response time of the memory to read-write requests was forcibly increased beyond the permit-

ted limit of 2.5 microseconds. This checked the Time Out circuitry in the Bus Controller logic which checks against undue delays in response by the memory or the peripherals.

In each case, the system was able to diagnose the faults.

6. Conclusion

One of the main objectives of this design effort was to establish that a high reliability processor can be designed from medium reliability components using microdiagnostic techniques. The present design met the projected performance specifications and the environmental specifications. Moreover, use of Bit Slice microprocessors and microprogramming techniques resulted in a very compact design. The field endurance tests of the mobile Electronic Telephone Switching System amply proved the suitability of this design approach for numerous applications in developing countries which require ruggedised processors. Tests carried out at the Cyclone Warning Radar Station at Madras, India proved the adequacy of the design for non-mobile data processing type of activity. The success of the processor for the digital switching application proved its reliability in the mobile role, thus establishing the versatility of the design.

7. Acknowledgements

Our thanks are due to the many colleagues who helped and assisted in implementing this system and the many organisations which participated in the environmental and field tests and evaluated its performance. The work reported here was funded by the Department of Electronics of the Government of India.

References

- [1] K.C. Anand, M.V. Pitke and P.V.S. Rao, 'R-1000, A general purpose medium size Digital Switching System for Developing Countries', *Int. Conf. on Communications (IEEE)* (1985, Chicago, 111) pp. 641-645.
- [2] F.J. Hackl and R.W. Shirk, 'An Integrated Approach to Automated Computer Maintenance', *IEEE Conf. Record*

- on Switching Circuit Theory and Logical Design* (1965) pp. 289-302.
- 3] R.W. Cook, W.H. Sisson, T.F. Storey and W.N. Toy, 'Design of a Self-checking Microprogram Control' *IEEE Trans. on Computers*, Vol. C-22, No. 3 (March 1973) pp. 255-262.
- [4] Am 2901 Bit Slice Microprocessor Handbook (1976, Advanced Micro Devices, USA).