EVALUATION OF THE CLARION COMPUTER PROPOSED TO THE NOXELL CORPORATION

April 14, 1986

IC Engineering, Inc.
P.O. Box 321
Owings Mills, MD 21117

(301) 363-8748

1. General

2. Hardware

3. Firmware
   - 3.1 Program Updates
   - 3.2 Diagnostics

4. Testing

5. Statistical Data Collection
   - 5.1 Hardware Modifications
   - 5.2 Firmware Modifications

6. Cost Estimate

7. Recommendation

1. General

The proposed Clarion Computer is a relatively straightforward application of contemporary microprocessor technology. As in many design areas, implementation choices are influenced by individual preference and experience. This includes the selection of particular components, architectural layout, hardware and software tradeoffs, system flexibility, and software design philosophy. There are a multitude of design choices which would satisfactorily meet the imposed constraints. The use of a single-chip microprocessor to reduce parts count and board size; selection of other processor chip families to eliminate clock generation circuitry or input/output devices; and alternate power-down schemes are some overall design judgements to be made early on in the development phase.

It is not the intention here to quibble over an individual engineer's preferences, but rather to evaluate the proposed system's design consistent with the following goals:

(1) is the overall architecture reasonable for the system requirements, or is there an overpowering reason to select a different design methodology?

(2) given the selected architecture, are any design flaws evident in the particular implementation?

(3) within the chosen framework, can improvements be made?

(4) what would be required to increase functionality?

Within these guidelines, the hardware, firmware, and proposed testing scheme are analyzed.

2. Hardware

The requirement for a long operational lifetime under battery operation mandates low power consumption. The selection of CMOS family logic is a natural consequence of this one constraint. Standard CMOS logic gates, CMOS memories, and a CMOS version of the Z80 processor are used in the Clarion Computer. The Z80 CPU is more than adequate both computationally and from a speed viewpoint to handle the intended task. However, it is not significantly more expensive than less powerful CPU's, and so is a reasonable choice.

Liquid crystal displays draw a small amount of operating current, so the use of these displays is an intelligent decision. The voltage regulator and charge pump circuitry used are specifically designed for low power consumption. The two potentiometers are provided to permit varying the voltage source to the LCD's. It would be desirable to eliminate these adjustments if possible: the likelihood of incorrect adjustment, and the lowering of reliability are detrimental effects. If it is not possible to operate from a single fixed voltage, the possibility of using two or three fixed voltages, selected either by shorting clips or dipswitches, should be explored. Upon discussion with the manufacturer, it was learned that the adjustments are used to compensate for variation in manufacturing lots of the displays. Once adjusted at the factory, there would be no need for readjustment. The manufacturer is working on eliminating the adjustments; if this cannot be done, a smaller trimpot which can be painted to lock subsequent drift would be preferable.

Final selection of the keypad should be carefully made to ensure the long time survivability of the switches. The keypad is the one portion of the entire system which must accept repeated abuse from customer usage. Elastomer type switches can be susceptible to fingernail gouging, etc., and should be tested for vulnerability. Insulation to high voltage potentials should be tested, for both dielectric breakdown and induced voltages resulting from static discharges. Although the selection of CMOS logic is clearly indicated because of low power requirements, the CMOS family is most susceptible to static damage. Adequate protection from all outside sources must be provided; the keypad is the most likely entry point for high voltage discharges.

Depending upon the results of static discharge testing with the keypad, it may be desirable to include protection circuitry on the board from such induced voltages. A varistor surge absorber across the power supply leads, and diodes protecting the row and column lines from the keypad would be prudent. Two signal diodes, to Vcc and ground for each row and column lead, or a single zener diode for the row and column leads should prove adequate. Discussion with the manufacturer indicated that

similar modifications are being planned at this time.

Static testing of the LCD's should be performed, as well. It is less likely that a customer would touch the display than the keypad, however. Static damage to the displays would be more likely to result from a cleaning operation than from customer use. The manufacturer stated that future units will have a clear plastic shield isolating the displays from the customer -- this should protect the LCD's from cleaning chemicals. Testing should be performed to verify the quality of the seal.

Considering that the battery pack is capable of significant short circuit current, it would be desirable to place an inline miniature fuse in the cable connecting the battery. The voltage regulator has short circuit protection and is unlikely to permit an overcurrent condition through it; however, the fuse would serve to protect against shorts in the battery connector and circuitry powered from the Raw Supply source on the board.

The fewer sockets used, and the more parts which are soldered directly onto the board, the greater the reliability will be for the units, once delivered. The EPROM must use a socket to permit future software upgrades. The manufacturer plans to use sockets for the RAM and processor for the first thousand units; if satisfactory results are obtained with those, these parts will be soldered to the board. The disadvantage of not using sockets is that for the manufacturer, in the event of defective incoming components which are not detected prior to soldering, there is an increased cost for board repair. The manufacturer states that pretested chips are to be used, and sockets will be eliminated when initial experience demonstrates success.

3. Firmware

The control program is written in Z80 assembly language, and is contained in the 27C64 EPROM. The level of sophistication of the program is not particularly high. There are many areas of code which can be optimized to reduce code space and program complexity. However, this fact does not detract from the unit's proper functioning, and given that the device's operation has been verified to perform according to the desired requirements, does not affect the product negatively. It does make it more difficult to modify the program characteristics for future applications -- but the effort to redesign the code would be greater than the incremental difficulties encountered at modification stages. There is a possibility that rewriting the code would permit a smaller EPROM (27C32) to be used, at a smaller cost ($1 - $2), but it is impossible to ascertain this without undergoing the rewrite process.

The existing program fits easily (3/4 full) in a single 27C64. Considering that there is 25% room remaining, and that the code could be reduced if necessary, it is unlikely that a second EPROM will be needed. The only cost reduction which would be realized through the deletion of the second EPROM option is the cost of a socket, and a marginally lower cost printed circuit board. Since the board is already complete, there is no incentive to remove the second EPROM option.

3.1 Program Updates

Should the program characteristics desired change after product deployment, new programs can be distributed through the dissemination of replacement 27C64 EPROM's. The field replacement procedure would be to unplug the battery connector, remove the existing EPROM, and plug the new EPROM in. The original memories can be returned, erased, reprogrammed, and redistributed should yet another program change be required. Alternatively, a portion of the installed base could be updated, the returned chips reprogrammed, and redistributed in various stages until all had been replaced.

The difficulty of modifying the existing program to meet new requirements is surely dependent upon what those changes are. One observation to be made regarding the existing program is that the procedure of menu display and customer response is not done in a table driven fashion. Separate code exists for every question and response (although extensive use of common subroutines is naturally used). The program could have been written with the use of data structures stored in tables which would then define the actual customer interface. For a program coded using such means, customer interface options can be changed trivially through modification of the definition

transition tables, with virtually no alteration of the control program itself (and a concommitant lower likelihood of software bugs developing). This would also result in a smaller code space needed. This is not to say that future program changes will be exceedingly difficult; just that they will be more difficult than if the program had been written in a more rigorous fashion from the start.

3.2 Diagnostics

The existing program contains diagnostic routines which are used to test the unit. The diagnostics are accessed in two ways: by asserting a jumper option on the board, which is used in initial test by the manufacturer; and by depressing certain buttons on the keypad while resetting the processor. This permits execution of the diagnostics at any time after manufacture. The existing diagnostics test the RAM (6116) and the LCD displays. Improvements to these, and three new tests, are described.

The RAM test tests each location by writing the lower 8 bits of the location's address into each location, and reading back for verification. This is repeated with the complement value stored. This test does not test the address lines, either those on the printed circuit board, or those internally in the RAM. The same data are written into each page using this scheme; should an addressing problem not access different pages properly, this failure will not be detected. It would be possible for a single page of memory to be used over and over, and test successfully. A simple modification can be made which overcomes this deficiency. Simply add the higher 8 bits to the lower 8 bits of the location's address, and write this into the RAM. Repeat with the complement. Improper page addressing will be detected under this method, and only an instruction or two need be changed in the code in four places.

The LCD test activates all spots on the displays to verify that they darken. A more sophisticated test would be to rotate a pattern of dots through the display, so that failures consisting of improper addressing, either on the board, or internal to the displays, would be evident.

It is strongly recommended that a new test be added to verify the EPROM contents. Although it is assumed that the EPROM programmer used already reads back the data after programming, a self-contained test is useful to detect bits "reappearing" due to improper erasure, or other failure, and to test the addressing and data integrity of the assembled unit. The EPROM test can be quite simple, yet effective. An overall checksum of the entire EPROM can be easily calculated, and compared with the correct value. A go/no go result would be sufficient. Between

the combination of the RAM test and an EPROM checksum test, proper functioning of the address and data lines can be tested at any time.

A test to check the operation of the keypad could save some time in quality control. A section of code which simply displays the button which is depressed can be used to rapidly check each key.

An essential test should be added to check the operation of the watchdog timer. All this need do is halt the processor, or jump to the existing label, NOPOWER. If the timer is functioning properly, the computer will reset itself.

4. Testing

The testing methods planned by the manufacturer should prove satisfactory. The procedure should include current drain and voltage regulator output measurements at simulated high and low voltage limits of battery operation. Additionally, it is desirable to be able to perform as much functional testing with the final unit as is possible. This can typically be done by analyzing the circuit diagram and postulating a malfunction in each section, and providing a diagnostic routine accessible via the keypad which is capable of detecting the failure. The inclusion of an EPROM checksum test, a keypad test, a watchdog timer test, a shifting display test, and improvement of the RAM test should cover most of the circuit failure mechanisms. A bake and device burn-in may be desirable prior to final testing. In any event, a formal testing procedure worksheet should be developed covering automatic and manual testing procedures, showing that all potential failures will be detected at some stage of testing.

5. Statistical Data Collection

It would most likely be helpful to have the ability to collect statistical information related to the use of the Clarion Computer for use in developing customer profiles, correlation with sales, etc.

5.1 Hardware Modifications

In order to collect statistical data, provision must be made to store the information until the data can be retrieved. As a minimum, accumulators can be kept which simply count the number of responses of each type for every question. There is ample room in the 6116 RAM to store this information. However, power down of the unit must not perturb the accumulators. Several means can be used to handle this.

The power down procedure could be used to turn off the displays and the charge pump only, leaving the microprocessor and peripheral chips powered. This will draw more current, and result in reduced battery life. This method permits more sophisticated data collection, since the processor is active at all times. It would be possible to perform timing related data collection, such as: peak number of responses in a single half-hour, hour, or 12 hour time period; and real-time histograms of usage by time of day and day of week. These would be the more sophisticated of information collection plans, and additionally would require the provision of a real time clock. This could be done with the replacement of capacitor C2 with a crystal for timing accuracy.

Alternatively, the RAM could be powered at all times, with the rest of the circuitry powered down as currently done. If this method is used, provision must be made to safeguard the RAM's write line from erroneous activation during the power down, and power up sequences. This method could be used for accumulator collection, without time variable parameters.

An equivalent method to the previous one is to replace the 6116 with an equivalent device which maintains its own backup supply. Two such methods are: use of a Mostek MK48Z02 RAM chip; and the use of sockets (Dallas Semiconductor) which contain battery backup circuitry. Either of these two methods would require no changes to the circuit board. They each provide a RAM with voltage sensing circuitry and watch type battery backup for memory retention.

An alternative method would involve the addition of an EEPROM, with additional control circuitry. The existing 6116 cannot be replaced with an EEPROM, however, because it is used for temporary data values and the stack, both of which require

large number of write cycles; and EEPROM's are typically limited to 10,000 write cycles.

5.2 Firmware Modifications

In order to accumulate customer responses, memory locations need to be allocated for each question and answer. Upon receiving a response, the appropriate statistical accumulator must be incremented. A provision must be made to clear all accumulators initially, and a keypad sequence used to read the accumulators out. A continously increasing accumulation would be simplest to implement and utilize; the data could be read daily, weekly, or monthly, depending upon the refinement of the marketing information desired. The counters would operate much the same as an electric meter, so missing a reading would not cause loss of overall data, only the knowledge of the breakdown in time.

The additional code required to implement the above function would probably fit in the single EPROM; certainly rewriting the code to use state transitions would facilitate this upgrade. Alternatively, the additional code could be written and a second EPROM installed.

6. Cost Estimate

This section is included only to provide a rough estimate of the cost to the manufacturer for the Clarion Computer. Inquiries should be directed to the manufacturer for accurate information.

Integrated circuits:	$ 18
Small parts:	4
Battery:	6
Keypad:	3
Housing:	15
PC board:	6
Assembly/testing:	20
LCD's:	65
	$ 137

Typical multiples are between 2 and 5 times cost for retail prices. The above estimates would then show an expected cost of $275 to $685. Under normal situations, the customer would not be paying for setup fees; this should be viewed against the markup used by the manufacturer, which appears to be on the low side of the range. Development work already completed may or may not be applicable to other products for the manufacturer.

7. Recommendation

The Clarion Computer is an intelligent design using state of the art components. The software design is not as clear cut as it could be. This might increase future modification costs, but does not detract from its present operation. It is not recommended that basic software redesign be done, since it meets all current specifications.

When an evaluation unit is available which closely approximates the production units, several tests should be performed: vibration, heat, humidity, keypad abuse, and static discharge application. Depending upon the results of these tests, component or circuitry modifications may be indicated. The manufacturer states that FCC radiation testing will be performed. All of these tests will be most useful only if the unit tested is the final version. In the interest of minimizing delivery time, these tests may be performed prior to receiving a final device; however, if the end model differs from that originally tested, the tests should be performed again for verification.

The suggestions regarding the addition of a fuse, improvement of existing diagnostic routines, and addition of new diagnostic routines, described in prior sections, are recommended. The code changes for RAM testing, EPROM checksum verification, and watchdog timer testing are minimal and should definitely be made.

Modifications of the program to incorporate statistical data collection should be pursued. Implementation of the minimal accumulators in firmware should cost the manufacturer on the order of 40 hours of programming time. It is recommended that this be investigated, so that if only on a spot basis, particular units can be equipped with backup memories to utilize the statistical software.

The pricing specified by the manufacturer appears to be reasonable for an arms length transaction. Special considerations as a result of combining marketing studies with technical development can only be evaluated by the customer's organization.