Use of a micro-computer in a missile simulator

by JAMES A. ROSE and JAMES V. LEONARD
McDonnell-Douglas Astronautics Company-East
St. Louis, Missouri
and
HERBERT A. CROSBY
University of Missouri
Rolla, Missouri

INTRODUCTION

In 1971 McDonnell Douglas Corporation was awarded a contract by the Department of the Navy to develop a new weapon system. The weapon system is comprised of an advanced missile, several Control and Launch Subsystems (CLS) and the support equipment required to test and maintain the missile and CLS.

In a complex advanced missile system the need for some type of missile simulator is obvious. Missile simulators can range in complexity from small, rather simple, manually operated, limited capability devices to major pieces of automatic test equipment capable of simulating the missile and isolating CLS malfunctions to the lowest replaceable component level.

The primary requirement for a missile simulator is to “look like” a missile so that all the Launch System circuitry can be verified quickly and safely. Another desirable characteristic is to provide the capability of simulating a missile “NO-GO” condition to the CLS so that CLS operators can be trained to respond properly to possibly dangerous missile or launcher malfunctions.

MISSILE SYSTEM DESCRIPTION

The types of missile systems which interface directly with the CLS, and therefore must be simulated are, (1) the electrical power system, (2) the discrete command/response system and (3) the digital command/response system. The power system consists of 400 and 60 Hz AC power circuits and several 26 or 28 v-de circuits. The discrete command/response system consists of the circuits required to process individual 26 v-de signals which activate missile subsystems and generate the discrete on/off responses which indicate system activation or status. The digital command/response system consists of the digital circuits required to receive and process information in digital form. This information is used to update the missile guidance system and to control all the automatic functions required to launch the missile rapidly and accurately.

The missile digital system interfaces directly with the Missile Data Processor (MDP) in the CLS. The MDP is a digital computer with an 8,000 word memory capability. Its output to the missile consists of three distinct signal types. These are (1) Data Enable, (2) Clock strobe and (3) Digital Data. The timing diagrams for data transmission are shown in Figure 1 and are based on the MDP specification per Reference 2.

Because of the 100 kilobit/second rate of data transmission, a method to verify the accuracy of the received information is required. In this system, odd parity and checksum methods are used to validate all transmitted digital data.

The missile system is capable of producing only two responses to the MDP. These are the “Good Data” word and the “Missile Status” word. If a checksum comparison exists and the parity is valid a “Good Data” response is returned to the MDP. Upon command from the MDP the missile runs through a built-in-test (BIT) routine and responds with a “Missile Status” word. Each bit is the GO/NO-GO status of a particular missile system or subsystem.

SIMULATOR DESIGN CRITERIA

Early in 1973 the requirement for an improved missile simulator was recognized and a contract was given to design and build a limited number of development test units. The new simulator was to be called the Missile Simulator/Test Set (S/TS). Prior to starting the detailed design of the S/TS, a set of design criteria were established utilizing the most desirable characteristics of the previous simulator and doing away with the least desirable characteristics. These are:

1. The S/TS shall be portable.
2. The S/TS shall be a self-contained unit requiring no external power source.
3. The S/TS shall have a load bank circuit which simulates the missile electrical loads on a time shared basis.

4. The S/TS shall be capable of automatic response to selected discrete commands.

5. The S/TS shall be capable of providing a "NO-GO" condition on the discrete output lines by inhibiting the automatic command response at the operator's discretion.

6. The S/TS shall be capable of automatic response to all digital inputs.

7. The S/TS shall have circuitry to validate incoming digital data utilizing the parity and checksum methods.

8. The S/TS shall be capable of simulated "NO-GO" digital responses to the MDP in the following manner:
   (a) by inhibiting the "Good Data" word output at the operator's discretion.
   (b) by simulating a missile subsystem NO-GO condition in the "Status" word at the operator's discretion.

9. The S/TS digital system receivers, drivers, and timing shall be identical to that of the missile digital system.

10. The S/TS shall have a built-in-test (BIT) capability to test the internal loadbank, discrete and digital circuitry.

The S/TS Control Panel is shown in Figure 2.

POWER SYSTEM SIMULATION

In simulating the missile power loads, the S/TS does not use the conventional fixed value, high power dissipation, resistor load approach. It was considered adequate to load the CLS power output circuitry periodically for short spans of time to verify its capability to supply the required current at the stipulated voltage levels. This allows derating the load resistor power ratings as much as three orders of magnitude, depending on required sample duty cycle. This in turn greatly reduces the load bank size and heat dissipation requirements but increases its complexity. It was decided that size and weight were of major concern, hence this approach was adopted for the S/TS.

DISCRETE SYSTEM SIMULATION

In order to alleviate the necessity for a simulator operator to be present for every test, the S/TS was designed to automatically generate the required responses to all discrete commands. This is accomplished by simulating the missile relay loads with electronic relays which are used to control
activation of the response signal. Each discrete response
can be faulted (i.e., inhibited) by depressing a panel switch
indicator. This approach is somewhat more complicated
and the hardware more expensive than that of the original
simulator but was considered a better simulation of the
missile discrete command and response system.

DIGITAL SYSTEM SIMULATION

The shortcomings of the original simulator digital com­
mand/response system were eliminated by a complete
redesign of the digital section for the S/TS. The basis of
the new design is the MCS-4 micro-computer set with
additional 256 word, programmable - read - only - memory
(PROM). A block diagram of the S/TS digital section is
shown in Figure 3.

There is a considerable amount of peripheral logic cir­
cuity required to perform the input/output functions for
the micro-computer set. One of the first steps in designing
the peripheral logic and generating a workable program for
the micro-computer is to prepare a logic tree and from
that, a flow chart.

Because of certain format characteristics of the input
command words, the logic tree can be made rather simple.
(Reference Figure 4.) Each input command word is 16 bits
in length and is to be analyzed in bytes of 4 bits each. (Byte
1 is the least significant 4 bits.) The acceptable input com­
mand words are one of the following forms:

(a) (U000)16 Request MSL Status
(b) (0V00)16 Reinitiate
(c) (00WW)16 Reset I/O
(d) (00XY)16 Data Group Identifier.

NOTE:
(a) The above command word forms are shown using
base 16 format.
(b) For reasons of security, the actual values of “U16”,
“V16”, “W16”, “X16”, and “Y16” are not given.

LOGIC TREE DESCRIPTION

As can be seen from the above and in Figure 4, if byte 1
is zero, the received word must be of form a) or b) and,
byte 2 must also be zero or an input error exists. If byte 1
and 2 are both zero, only two possibilities exist for bytes 3
and 4; either byte 3 is zero and byte 4 has value “U16”, indi­
cating a Request BIT Status word has been received or byte
3 has value “Y16” and byte 4 is zero, indicating the Reinitiate
word has been received. All other possibilities are invalid
and result in a digital error.

If byte 1 is non-zero, the received word must be of form
c) or d), hence byte 2 must also be non-zero. If this is true,
then bytes 3 and 4 must be identically equal to zero or a
digital input error exists. If bytes 1 and 2 are both of value
“W16” a Reset I/O word input is indicated. This is because
there is no other valid command word which is identically
equal in value to (00WW)16. If byte 1 and/or byte 2 is a dif­
ferent value than “W16”, a Data Group Identifier word of form
(00XY)16 is indicated. The value “Y16” for byte 1 is a code
indicating a specific block of data is being transferred and
the value “X16” for byte 2 is the number of words to follow,
including the checksum word.

FLOW CHART DESCRIPTION

The following is a description of the S/TS Digital System
Flow Chart with explanations as to how some of the func­
tions are to be implemented. As is seen in Figure 5, there
are four data processor “modes” mentioned. These are
defined as follows:

Mode 1. The data processor program is "set" to expect, and accept, only the "Reset I/O" command word; i.e., (00WW)16, which is required to initialize the system. Any other input word will indicate a Digital Error (Note: the "Reset I/O" command is required prior to every input command word).

Mode 2. A "Reset I/O" has been received and the Data processor is "set" to expect any command word. (Including "Reset I/O")

Mode 3. A valid command word has been received. If it is of form (00XY)16, set the data processor to expect X=1 data word.

Mode 4. The last data word (X=1) has been received, the data processor is set to expect the checksum word as the next transmission.

In Figure 5 it can be seen that initially the processor is set to mode 1. A Data Input Enable command is generated in the Processor Output Command Channel. The system remains in this state waiting for a "Data Enable" from the MDP. With receipt of a "Data Enable," the accompanying input data is clocked serially, least significant bit (LSB) first, into the Input Buffer (I/O register). As the first four bits enter the register, they are read out in parallel "on-the-fly." If byte 1 does not equal zero, it is stored and bytes 2 and 3 are read and stored. (Reading the data in bytes as it is being clocked in instead of storing each byte and then going back to recall it, read it, and re-store it saves considerable processing time and I/O hardware.)

The fact that byte 1 does not equal zero indicates that the incoming word is a "Reset I/O" command word, (00WW)16, a "Data Group Identifier" word, (00XY)16, or some invalid command word. The "Reset I/O" and "Identifier" words both require that bytes 3 and 4 be equal to zero. If byte 3 is zero, byte 4 is read and stored. If byte 4 is also zero a parity check is made of the word. If the parity is even a Digital Error exists. If it is odd, byte 2 is recalled and read. If byte 2 is non-zero, bytes 1 and 2 are then checked to see if they each equal "W." If so, a Good Data word response is generated and clocked out to the MDP. The S/TS processor program is then set to mode 2. If bytes 1 and/or 2 are not equal to "W," the processor mode would be checked. If the processor is in mode 1; i.e., expecting a "Reset I/O" command, a digital error is indicated. If the word preceding the incoming command word had been a valid "Reset I/O" command, the processor would now be in mode 2, hence the incoming word must be a "Data Group Identifier" word.

The "Identifier" word "Y" value is evaluated and compared to a listing of acceptable Y values. The "X" value is evaluated and a Data Word Index Counter is initialized to the value (X=1). A partial checksum consisting of the "value" of the Identifier word is jam transferred from the I/O register into a parallel adder. (Note: this type of adder is such that each sum is retained and the next addend is added to it, keeping a running total). After the Data Word Counter is initialized and the partial checksum loaded the processor program is set to mode 3.

Assume that the next input word received has byte 1 equal to zero. Since the processor is not in mode 1 or 2, the received word is assumed to be a data word and all four bytes are read "on-the-fly" and stored in the I/O register. Odd parity of the input word is verified and the processor mode is rechecked. If even parity exists or the processor had been in mode 1 or 2, a digital error is indicated. With the processor in mode 3, the Data Word Counter is decremented by one, and the data word "value" is jam transferred into the checksum adder. If the Data Word Counter has not been decremented to zero, the last data word has not been received and the processor remains in mode 3.

The remaining data words in the Data Group can be of any "value" hence, bytes 1, 2, 3, and 4 may or may not equal zero. If byte 1 does not equal zero the same path in the Flow Chart as was followed for the "Reset I/O" or "Identifier" word inputs is again followed. The difference is that if bytes 3 or 4 do not equal zero, or byte 2 does equal zero, the logic flow is laterally transferred to a like functional location on the path followed by the first Data Word.

When the Data Word Counter is decremented to zero, the last data word has been received and the processor program is set to mode 4. The next word received, therefore, should be the checksum word. The checksum can, obviously have any "value," hence can follow any one of the logic flow paths described previously, eventually getting to the mode 3 check in the lower center of the flow chart. Since the processor is no longer in mode 3, a checksum comparison is made. To make the checksum comparison the received checksum is clocked into the I/O Register, the register is complemented and the checksum complement is jammed into the adder. If the adder sum is all "1s" a valid checksum is indicated and a "Good Data" response is generated and clocked out to the MDP.

Assume byte 1 equaled zero and the processor was in mode 2. It is obvious that only two valid command words exist under these conditions. These are the "Reinitiate" command, (0V00)16, and the "Request Bit Status" word, (U000)16. For both of these words, byte 2 must also equal zero or a digital error is indicated. If byte 3 equals zero, byte 4 must equal "U" or a digital error exists. Parity is checked on the received word and if it is valid the "BIT Status" word is generated and clocked out to the MDP.

If byte 3 does not equal zero, then byte 4 must equal zero or a digital error is indicated. After the parity check, byte 3 is compared to the value "V." If they are equal a "Good Data" word is generated, a 50 msec Input Data Lockout is initiated and the processor is reset to mode 1 to await a "Reset I/O" command.

The digital system design and micro-computer program dictated by the Logic Flow has been simulated on the McDonnell-Douglas Automation Company SIGMA 47 computer. The chart just discussed is the result of two previous flow chart simulations which pointed out several
Timing and sequencing problems. Use of the micro-computer and the results of the simulations make the S/TS Engineers confident the unit has been designed to make maximum use of available hardware and will accurately simulate the missile.

SUMMARY

In 1971 McDonnell-Douglas was awarded a contract for a new missile system. The missile power, discrete command/response, and digital systems were described with emphasis on the digital data format, error detection and timing. A missile simulator, the S/TS, was briefly described showing how the shortcomings of a previous simulator were eliminated. A listing of the S/TS design criteria was given and the design logic based on utilization of a micro-computer was described.

ACKNOWLEDGMENTS

The authors wish to express their appreciation to Mr. Bruce V. Savage and William G. Kull, who were given the detailed design responsibility for the S/TS circuitry and microprogramming, for their advice in preparing the figures used in this paper. Also the authors wish to recognize Mr. James E. Moore of McDonnell Douglas and Mr. Lowell Noxin of the Naval Air Systems Command for their help in establishing the basic design criteria for the S/TS.

REFERENCES
