BELLCOMM, INC.
1100 Seventeenth Street, N.W. Washington, D. C. 20036

SUBJECT: Organization of Half Size AGC Computer Program - Case 310

DATE: March 27, 1968

FROM: W. G. Heffron

MEMORANDUM FOR FILE

Introduction

The short feasibility study recently conducted by Bellcomm, Inc. for the Apollo Guidance Software Task Force, NASA, had two goals. One was to use only half the available memory, the other to use only one interrupt. This memorandum reports on the one interrupt part of the study.

By way of definition, an interrupt is a signal to the computer to stop activity in a routine, branch to another routine, perform necessary computations and then return to the original routine.

Present or Interrupt Structure of the AGC

Figure 1 notes that the Apollo Guidance Computer has two types of interrupts, which are distinctly different.

Counter Interrupts are special hardware features of the AGC. As an example, when an integrating accelerometer outputs a pulse representing some 0.1 fps sensed velocity change, a counter interrupt occurs which changes the contents of an AGC memory work by one count. Later, when programmed to do so, the AGC routine can examine this word to determine how much velocity change has been sensed. Counter interrupts are a hardware feature, not a software feature, and involve the computer only in that a memory word is used and in that one machine cycle is used to bump the word content by one count when the interrupt occurs. Programmers do not need to know about counter interrupts.

The alternative to a counter interrupt is an accumulating register external to the AGC, which is read (serially, usually) by the AGC when programmed to do so. The data is then transferred into an AGC memory word. This alternative thus requires as much AGC storage if not more, as much time if not more (especially for serial transfer), and additional hardware. It was concluded, therefore, that these interrupts should be used and be considered only as a hardware feature of the AGC, not as true int—

Program Interrupts are signals, response to which must be programmed. For example, to turn the engine off at a precise time, the AGC software sets an internal clock to a value equal to the time-to-go. The clock counts down to zero, and zero causes an interrupt. The program branches to a location in memory uniquely associated with this particular interrupt, determines that "engine off" is the proper response, issues the command, and then returns to the original routine in progress before the interrupt.

In some cases, it is understood that the interrupt routine does even less than this before returning to the routine in progress. Instead of actually doing the special computations, the task is entered into the waitlist (see next paragraph) as a job waiting to be done.

Present AGC program design depends greatly on these program interrupts. Various tasks are assigned priority levels from 1 to 32 and at frequent convenient times, the software routine in progress is changed, becoming the one having the current greatest priority (until a higher priority job enters the queue). For example, when a specific task, e.g. navigation, is to begin two seconds later, this time is entered into a waitlist, an interrupt occurs two seconds later and the navigation is executed as soon as its priority is the greatest in the then current queue of tasks awaiting execution.

Timing Demands

An alternative program organization is to check frequently enough if it is time to do something, rather than doing it at "exactly" the required time. Timing demands are shown in Figure 2, and are the basis for the discussion which follows.

The ability to turn RCS jets off with a time granularity of 1 ms is highly desirable because fuel can be saved in that way. But checking every 1 ms to see if it is time to do so takes too much time from other activities. This function is then the task which must be kept as an interrupt.

Down telemetry can accept new data (two words) every 20 ms. The rate used to be slower, and this speed was recognized as necessary and implemented. It should therefore be kept, although it is somewhat more often than desirable from a programming viewpoint. And the return to the telemetry routines should be fairly synchronous at this rate.

Sextant marks should be recognized and serviced within some 40 ms of their occurrance, if the sextant data is to be useful in lunar landmark tracking.

The Digital Autopilot Routines require new computations about every 60 ms. Service should be fairly synchronous, since the digital filters in the autopilot depend upon a constant time interval between gimbal angle readings.

DSKY (Display and Keyboard) entries and hand controls should be serviced within some 100 ms of their occurrance, if crew members are not to notice a delay in response to their actions.

Uplink telemetry should be serviced within about 100 ms of the arrival of the uplink information. Communication system performance, not the AGC, determines this rate.

The G & N housekeeping cycle is about 100 ms and should be maintained. This includes such operations as Inertial measuring unit status check, temperature checks, accelerometer failure tests, etc., although not all tests are done every 100 ms.

Alternative Program Organization

These program interrupts have a hardware basis as well as a software basis. To avoid changing hardware, the software is changed so that a program interrupt posts a flag indicating the need to do a certain task.

The one interrupt allowed is used as a timer interrupt having two purposes--to turn RCS jets off and to initiate the 20 ms basic cycle, as is shown in Figure 3.

Down telemetry is the first part of this 20 ms cycle--every 20 ms two new words are transferred to the output channels.

The digital autopilot 60 ms cycle job follows immediately after the telemetry operations because this maintains the cycle of 60 ms best. Autopilot calculations are begun on the first, fourth, seventh, etc. 20 ms cycle and one-third of the autopilot is done each 20 ms cycle.

Sextant marks (40 ms service rate) come next. Here 40 ms means that the flag should be checked at least every 40 ms. Actually once the mark is made it is usually several seconds before a second mark occurs, so servicing the mark can be permitted to take more than 40 ms. But the servicing job must be divided into many parts each of which fits into the 20 ms cycle.

The other jobs fit into the 20 ms cycle in much the same way--each must be divided into parts which can fit into the 20 ms cycle, and the sum of the parts must in no case exceed 20 ms. For the general situation in which less than 20 ms is required, the idle period shown is included, and the exact 20 ms cycle insured thereby.

In this organization the executive has two levels. The high level one switches from 20 ms to 60 to 40 ms, etc., jobs. At the lower level, the pieces of the 100 ms (and 60 and 40 ms) tasks are connected together into a meaningful sequence. This overhead is estimated to cost about 1% of machine time: the idle overhead is estimated at about 10% (based on the LM Abort Guidance System experience).

Thus under this organization there is a forced idle period, which means that the AGC must be somewhat slower than it is at present. Under the present scheme the AGC is idle only when there is nothing to do. Because of this, it is likely that the slow cycle for guidance would increase to somewhat more than two seconds, estimates being three seconds or less. This should be acceptable, however.

Variations

Variations on this scheme are easily conceived and two are now presented.

One is to add the 20 ms down telemetry to the timer interrupt (giving it three functions instead of two) and set the basic cycle to, say, 90 or 100 ms. The 60 ms job would become a 45 or 50 ms job appearing twice in the 100 ms cycle, while the 40 ms would be either 30, 45, or 50 ms as would be most appropriate. This reduces the overhead somewhat, particularly the idle portion.

Another is to increase the storage capacity of the telemetry so it can hold 10 words (100 ms worth). Then telemetry would not be an interrupt and the 100 ms cycle scheme above could be used.

Conclusions

It is possible to reorganize the AGC computer program to reduce the number of programmable interrupts to only one--a timer interrupt used to turn RCS jets off to within 1 ms granularity and to control a basic 20 ms cycle of operations.

No hardware changes are required for this approach: present counter interrupts should be kept and used, and present program interrupts used to set flags indicating that servicing is required.

Many variations on this basic approach are possible.

2014-WGH-bjh

W. G. Heffron

29 COUNTER INTERRUPTS - NOT PROGRAMMABLE, REQUIRE 1 MACHINE CYCLE TIME (12 μ SEC.), NOT UNDER COMPUTER PROGRAM CONTROL, CANNOT BE INHIBITED. EXAMPLES: PIPA COUNTS, CDU PULSES

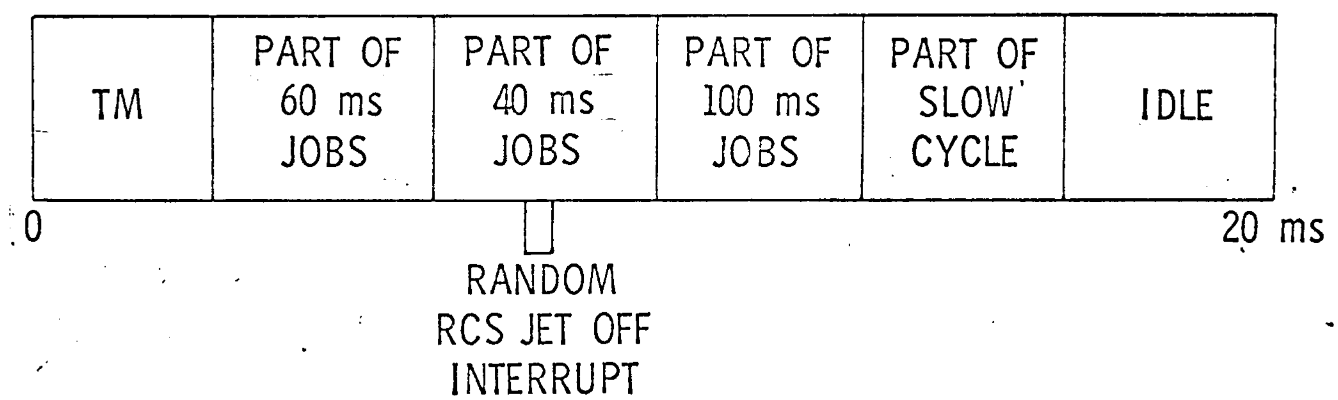
11 PROGRAM INTERRUPTS - 4 CLOCKS, 2 DSKY, 1 UPLINK, 1 DOWNLINK, 1 RADAR, 1 HAND CONTROLLER, 1 RESTART. ALL BUT RESTART CAN BE INHIBITED, BUT ONLY AS A GROUP. SERVICING AN INTERRUPT REQUIRES A MINIMUM OF 4 MACHINE CYCLES (46.8 μ S)

FIGURE 1 - INTERRUPT STRUCTURE OF AGC

- RCS JET OFF COMMANDS AT GRANULARITY OF 1 ms
- DOWN TELEMETRY NEEDS NEW DATA EVERY 20 ms
- SEXTANT MARKS NEED CHECKING ABOUT EVERY 40 ms
- DSKY, HAND CONTROLS NEED CHECKING ABOUT EVERY 100 ms
- DAP CYCLE TIME ABOUT 60 ms
- HOUSEKEEPING CYCLE IS 100 ms
- GUIDANCE, NAVIGATION CYCLE ABOUT 2 SEC. IN POWERED FLIGHT, SLOWER IN COASTING FLIGHT
- UPLINK GIVES NEW DATA EVERY 100 ms

FIGURE 2 - TIMING DEMANDS

- ENDLESS PROGRAM CYCLE REPEATS EVERY 20 ms ON TIMER INTERRUPT
- TIMER ALSO USED FOR RCS JET OFF INTERRUPT



- ABILITY TO STORE TELEMETRY DATA FOR NEXT 100 ms VERY DESIRABLE HARDWARE CHANGE

FIGURE 3 - PROGRAM ORGANIZATION