MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Instrumentation Laboratory
Cambridge 42, Massachusetts

DANCE Memo #2, COLOSSUS Memo #1, LUMINARY Memo #1

MEMO TO: Distribution

FROM: John R. Rhode

SUBJECT: Nothing's Here to Stay - Not Even $K\bar{K}K$

DATE: November 7, 1967

The computer uplink interface consists of a "ones" and a "zeros" line which feed information into a special uplink interface register. When a pulse is transmitted on the "ones" line, the contents of the register are shifted to the left one place and a one is introduced in the least significant bit. If a pulse is transmitted on the "zeros" line, the contents of the uplink register are shifted to the left one place and a zero is introduced in the least significant bit. An uplink interrupt is generated in the computer when a one is shifted out of the most significant bit (15th bit) of the uplink register (overflow of the uplink register). When the uplink is processed, the contents of the uplink register are read and replaced by zeros.

To protect the computer uplink interface against noise the uplink commands utilize an error detection code. The basic information received via the uplink is represented by a five bit keycode. These codes have the same bit configuration as the five bit keycodes which come from the DSKY allowing processing by a common routine (CHARIN). The error detection code is $K\bar{K}K$ where K stands for the five bit keycode and $\bar{K}$ stands for the ones complement of that code.

Before discussing the efficiency of the $K\bar{K}K$ error detection code something should be known about the most probable error source. Noise appearing on the ones and zeros lines at the computer interface can be interpreted as data transmission and cause spurious bits to be entered in the uplink register. This register should be in its zero state awaiting data transfer most of the time. At the maximum uplink transmission rate allowed by the ground, the time required to shift the data into the uplink register is 16 mils out of the total 160 mils required to transmit the data word to the spacecraft. During maximum transmission rate, therefore, the filling of the uplink register to overflow required only 10% of the total time. If the total mission time is considered, this percentage is reduced by several orders of magnitude. Therefore, the most probable error source is noise bits which are introduced into the uplink register between valid data shifts.

If spurious noise bits were in the uplink register prior to shifting the data bits, premature overflow of the uplink register would occur. The uplink message which is sent across the computer interface is as follows:

bit no.	16	15 - 11	10 - 6	5 - 1
	1	K	$\bar{K}$	K

The leading one is sent to trigger the uplink interrupt by overflowing the uplink register. For purposes of further discussion the five bits in K will be represented by the letters a b c d e. Using this representation then, the contents of the uplink register at interrupt for the no noise condition would be

bit no.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	a	b	c	d	e	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	a	b	c	d	e

where the bars above the letters (a, b, etc.) indicate the ones complement of those bits. If bit one of the uplink register had a one noise bit in it prior to the transfer of data to the computer, the contents of the uplink register at interrupt would be

bit no.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	1	a	b	c	d	e	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	a	b	c	d

If e, the last bit in the keycode, were a zero, this bit format would pass the $K\bar{K}K$ test with the computer assuming the K = 1 a b c d. In general, if the bits introduced in the beginning of the uplink register (bits 15, 14, etc.) are the ones complement of the least significant bits of K, the contents of the uplink register will pass the $K\bar{K}K$ test and an erroneous K will be transferred to the DSKY routines for processing. "Triple redundancy" is not gained because of the symmetry of the $K\bar{K}K$ code (i.e., the bits shifted into 5, 4, etc. are the complement of the bits shifted into 10, 9, etc. in the original and final formats).

A simple change to the on-board computer program would greatly enhance the error detection on the uplink register. If the code were changed to $\bar{K}\bar{K}K$, for example, only one line of coding on board would need to be changed and no change to processing time would occur. With $\bar{K}\bar{K}K$ the word format for no noise would be

bit no.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	a	b	c	d	e

If bit one of the uplink register had a one noise bit in it before data transfer to the computer, the contents of the uplink register at interrupt would be

bit no.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	1	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	a	b	c	d

In this case the bits shifted into bit positions 10 and 5 of the uplink register are the same but the format calls for them to be the complement of each other. In general no valid code may be shifted into another valid code with less than 10 noise bits present in the uplink register before the data shift. An example of 9 noise bits will illustrate this. The contents of the uplink register for 9 noise bits would be:

bit no.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	X	X	X	X	X	X	X	X	1	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{a}$

where Xs represent noise bits. Since bit 6 equals bit 1 and the $\bar{K}\bar{K}K$ format calls for these bits to be different, this error would be detected by the newly proposed code. The change of codes from $K\bar{K}K$ to $\bar{K}\bar{K}K$ would therefore greatly reduce the probability of incorrect data being accepted by the computer.

$\bar{K}\bar{K}K$ was chosen as the code since it would require only a one line on board program change and contains the keycode uncomplemented in bits 1 thru 5.

John R. Rhode

JRR/do