Saltzer

December 12, 1967

TO: Robert M. Fano

FROM: F. J. Corbató

SUBJECT: The effect of hardware reliability on software debugging.

The following memorandum gives a brief analysis of the debugging of Phase I for the Multics system on the GE 645 installation of Project MAC. The Phase I effort was selected as a particularly appropriate example since it has been given top priority for access to the machine and, as a result, represents the best possible effort in the face of all difficulties. The period selected is the final week of debugging before Phase I was achieved and it is considered fairly representative of the typical kinds of trouble which have been encountered for the last many months. (Simultaneously, but without as much priority, there were several other checkpoints being debugged in parallel.) It is characteristic of such a sample as this that it is not exactly the same as any other sample and, therefore, specific details should not be taken too seriously but rather the overall pattern and trends are the meaningful results.

The basis of the information for the analysis has been the software log maintained by the programming staff. The period covered is from 2300 November 29, 1967 to 0100 December 6, 1967. In an appendix to this memo will be found the detailed runs by date and time of initiation with brief comments as to their significance. This material is also

presented in summary form graphically in Figure 1. Overall, there were 57 accesses to the machine which break down to the following categories:

Useful for Phase I software debugging	8
Successful Multics system tape generation (MST gen) to support Phase I runs	13
Confusion in either run set-up or operation	7
Runs unnecessary by virtue of hardware malfunction	17
Runs unnecessary by virtue of hardware design errors	12
TOTAL	57

Of the above runs that were useful for software debugging, each of these required 30 to 70 minutes of machine time. The MST generation runs required about 10 minutes each when successful. The final Phase I run required 100 minutes. Thus it is seen that the sample under study represents a significant fraction of the machine's resources for the period in question. To see the effect of unreliable hardware, it is useful to see the breakdown of the time spend on the various aspects of debugging.

The 168 hours of the week break down approximately as follows:

Working hours (5x16 + 2x8)	96
Sleep plus rest (5x8 + 2x16)	72
TOTAL	168 hours

Of the 168 hours in the week, these can be broken down as follows:

Total hours in the week	168
Less time lost for hardware design errors	- 23
Subtotal	145

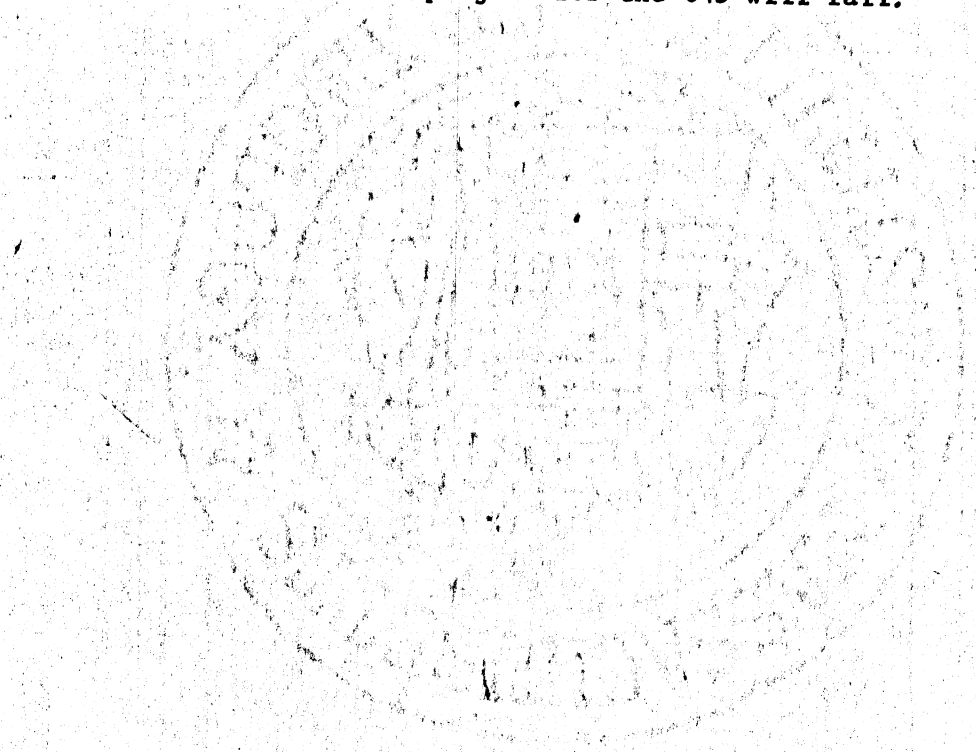
(Continued)

Less time lost for hardware malfunction	- 38
Gross time for working, sleeping plus rest	107
Less time for sleep plus rest (107x72/168)	- 46
Net useful software debugging time	61 hours

It should be observed that during the above week of calendar time there were only seven software bugs found in the Multics system programs that were being checked out. Thus it took on the average approximately nine hours of working debugging time to find each bug. It is especially important to note that with such long and difficulty debugging runs requiring much analysis and study that it is very easy to become discouraged and decide to give it up for the day (or the night) and go home. This breaking of momentum due to the malfunctioning of the equipment is an unknown price which is included in the above figures. It should also be noted that the design trouble which is mentioned in the appendix and which only appears for one day in the present sample existed for several days earlier and caused gross confusion in the proceeding week. Thus, if anything, the present sample is conservative in that there are no examples of hardware design troubles which last for several days.

In conclusion, it is possible to make the following observations. If there had been no hardware trouble to speak of, the entire week of work would have shrunk to five and one-half days. If these figures are extrapolated to seven months to develop Prototype Multics, then one has saved one and one-half months of check-out time. It is felt that this saving is a conservative one because of the momentum factor mentioned previously. Furthermore, if the hardware design errors had been caught

earlier, say by more careful diagnostic techniques during manufacture or more careful design, it is clear that there would be a corresponding one-day-out-of-seven speed-up in check-out. Finally, it cannot be over-emphasized that enormous energies are required of the programmers to diagnose hardware malfunctions and erratic behavior. Hardware reliability is a major factor in project morale since without reliability every system programmer knows the Multics project for the 645 will fail.



APPENDIX

Phase I runs or supporting runs

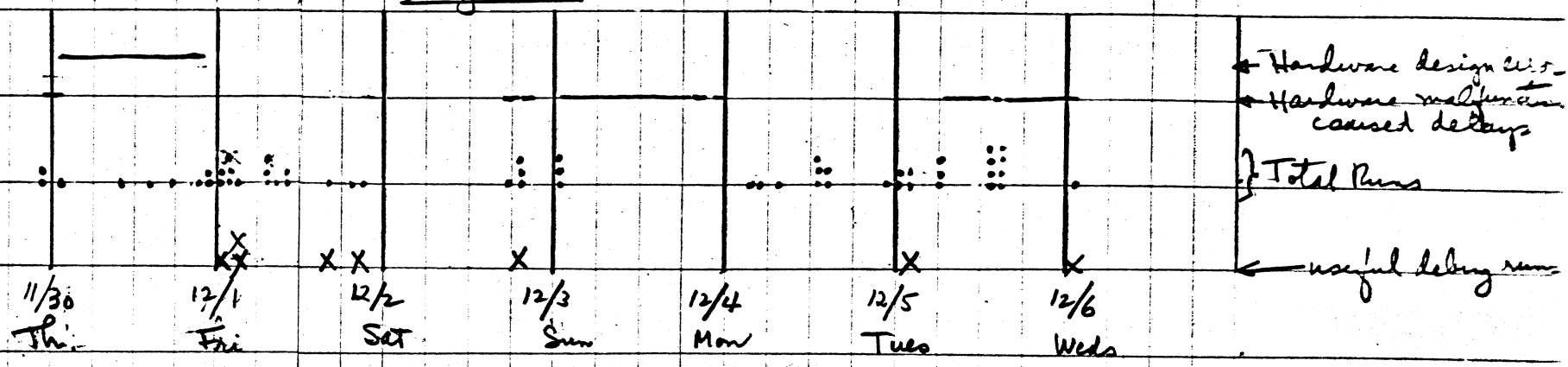
Key: Conf. = Confusion in either run set-up or operations
HM = Hardware malfunction
HDE = Hardware design error
Debug = Useful debugging run
MST gen = Multics System Tape generation

11/29	2315	Conf. - clocks misconfigured
	2330	HM - memory parity
11/30	0015	HDE - run would not go as far as previously
	1050	HDE - different stopping point
	1400	HDE - different stopping point
	1752	HDE - different stopping point
	2145	HDE - different stopping point
	2200	HDE - different stopping point
	2210	HDE - found hardware design bug
	2340	HDE - test to capture hardware design bug
12/1	0000	HDE - test to capture hardware design bug
	0015	MST gen
	0030	MST gen
	0040	Conf. - drum base address switches misset
	0045	Debug - bad rule in search module
	0145	HDE - test to capture hardware design bug
	0400	Debug - assembly error

12/1	0410	HDE - test to capture hardware design bug
	0420	MST gen
	0430	Debug - error in search rule
	0530	HDE - test to capture hardware design bug
	0540	MST gen
	1555	Debug - error in search rule
	1900	MST gen
	2100	Debug - generate_ptr spec faulty
12/2	1745	HM - bad MST gen
	1850	MST gen
	1900	HM - bad tape
	1915	Debug - echo not found
12/3	0015	MST gen
	0030	MST gen
	0040	HM - dump lost
12/4	1010	HM - teletype hung up
	1100	HM - teletype hung up
	1350	HM - did not repeat previous run
	1845	HM - bad MST gen
	1850	MST gen
	1905	HM - tape handler trouble
	1915	HM - operation not complete fault
	1930	Conf. - segment free_ too small
	2345	HM - bad MST gen
12/5	0000	Conf. - previous hardware design fix introduced another bug
	0030	MST gen

12/5	0200	Debug - mistake in get_removal_candidate
	0225	Conf. - mode change to Shell few days earlier which propagated
	0600	MST gen
	0615	HM - tape handler trouble
	0620	HM - tape handler trouble
	1300	MST gen
	1335	Conf. - wrong switch settings
	1340	HM - bad tape
	1350	HM - bad tape
	1425	Conf. - clocks on console not set
	1430	MST gen
	1440	HM - unexplained machine stop
12/6	0010	Debug - Successful Phase I

Figure 1



Key

• Run made

x Debug run