January 29, 1966

TO: Prof. F. J. Corbató

FROM: J. H. Saltzer

SUBJ: Status report on 645 Clock system.

On January 24, 1966, a techniqal proposal for a system clock by George Futas of GE in reply to repository document m0052, "A Proposed System of Clocks for Multics" was received at MAC and BTL. This techniqal proposal has been carefully reviewed by V. Vyssotsky and J. Ossanna for BTL, and myself for MAC. A consensus was reached on suggested changes to the GE proposal, and I communicated these proposals to Futas by telephone. He has agreed to the suggested changes, and has promised to produce a second draft of the proposal (no date specified.) The remainder of this document is a review of the telephone discussion, and the requested changes.

1. Relation of Clock Controller to rest of system.

The technical prpposal introduces a new system module, a Clock Controller, which is basically a stripped-down Memory Controller with no memory and only three execute-interrupt cells. Although it is our understanding that the Clock Controller is a temporary strategem to obtain a centrally located clock without waiting for redesign of the standard Memory Controller, George indicated that it is not considered proper for a technical proposal to comment on the temporary nature of any of its contents.

The existance of the Clock Controller, however, places several constraints on possible 645 system configurations; these are outlined here to make certain that they are not forgotten. First, the interrupt generated by the alarm clock

in the Clock Controller cannot be given arbitrary priority relative to other system interrupts, since all Controllers, whether Clock or Memory are required to have a fixed interrupt priority sequence, controller A highest, B next, etc. Second, the Clock Controller cannot be placed arbitrarily between two Memory Controllers in the sequence A,B,C, etc., since this would leave a memory addressing hole and make interlace impossible. Thus in a four-Memory Controller system, the memory controllers would have to placed on ports ABCD or on ports EFGH to obtain interlace; any Clock Controller(s) must take the remaining ports with whatever priority they have. Practically, the alternatives are that the alarm clock interrupt is either the lowest or highest priority system interrupt. Since the nature of the Traffic Controller for the Multics Supervisor requires that alarm clock interrupts not be lowest priority, it follows that they must be highest.

Thus the required configuration of a four-memory controller system is: Clock Controller on port A, Memory Controllers on B,C, or D, ports E,F,G, and H. George has indicated that such a configuration is completely acceptable in the light of the present 645 system design. When a clock is ordered, the system order should be reviewed to insure that this configuration is obtainable.

2. Masking of Clock Controller.

The technical proposal states that the interrupt cells in the Clock Controller are not maskable. While in most cases this would be an unacceptable restriction, in the light of the above configuration restrictions it is possible to provide

simple supervisor coding to make up for this deficiency. (Again, it is emphasized that the Clock Controller is an interim device; future clocks will be placed in memory controllers, and have adjustable priority and masking as do all other system interrupts.) The strategy used to handle wake-up interrupts is to pass through a ~~four~~ five-instruction sequence of code which merely sets an execute-interrupt cell in a standard memory controller. This latter execute-interrupt cell can have any desired priority and can be masked. It is possible to ~~take~~ accept the alarm clock interrupt at any time whatsoever, since it uses no common data base with any other procedure, and requires at most 20 machine cycles. It is also guaranteed that the alarm clock interrupt cell will not ~~come in~~ again continue to be set after the initial interrupt is taken. We may note that exactly the same strategy must be used with the processor interval timer which cannot be masked either.

The combination of the masking, priority, and restricted configuration masking and priority difficulties, and ~~restricted configuration~~ indicate that all possible haste should be considered in the matter of shoe-horning the clock module into a regular Memory Controller.

§3. Duplexing of Acconnting Clock Hardware.

Apparently there are at least two philosophical approaches to take toward reliability in a system clock: 1) keep the clock running and correct, even if the path between it and the system has failed, so when the path is restored, the clock does not need to be reset; and 2) attempt to keep a path from the system to some clock open at all times, even if this means switching between unsynchronized clocks. The plug for reliability

in the technical proposal
in my original proposal was taken to mean the first alternative ~~in the technical proposal~~; in fact the consensus of MAC and BTL is that the second alternative is preferable. In the revised technical proposal, there will only be one calendar clock module in a Clock Controller instead of two; reliability may be obtained by ordering two Clock Controllers complete with Calendar Clocks, and planning on reconfiguration in case of failure.

4. Alternate power sources.

The ~~original~~ technical proposal was oriented toward providing for continuous clock operation through a power failure, ~~rather than~~ although the features provided for this possibility can be used to provide continuous operation through a vacation in which M-G power is shut down. A change in emphasis here will indicate that the alternate source of power for the clock is basically just a wall plug independent of the system ~~MG-set~~ M-G set.

5. Access to Calendar Clock.

The original technical proposal restricts access to the Calendar Clock to Master Mode programs only; this restriction was primarily to minimize changes to the processor since a deleted master-mode only instruction (Read Memory File Protect Register) will be used for this operation. If it is possible, this instruction will be changed to slave access in the revised proposal.

6. Number of Ports on Calendar Clock Controller.

Only two ports (maximum) ~~are~~ were proposed for the Clock Controller. It seems wiser to allow for four ports since the relative timing of delivery of redesigned memory controllers and additional processors to BTL is completely unclear at this time. The

9. Power Down interrupt.

The technical proposal calls for a system interrupt to be generated by the Clock Controller if clock power fails. This feature will be removed, since it may interact unhelpfully with the processor power down facility, in the event of a general power failure.

revised proposal will specify a maximum of four ports on a Clock Controller.

7. Operator switches for setting clock.

For some undetermined reason, the number of bits of the Calendar Clock settable by the operator was changed from 36 in M52 to 26 in the technical proposal. This will be changed back to 36 in the revised propesal. The revision will specify that octal/rotary switches rather than binary toggles will be used for manual setting of the clock. (36 bits allow setting the clock to the nearest 0.1 second, 26 to the nearest 67 seconds.)

8. General Specification Tightening.

George agreed to spell out some items in more detail, for example, the nature of the alternate frequency input to the clock (logic pulse or low level RF), the method of rate control on the internal oscillator, and the nature of the alternate DC supply, ~~for some cases.~~

→ Insert #9

9. Processor Execution Timer.

10

The technical proposal rewires the processor execution timer to count memory accesses rather than real time passage; however, page and descriptor table accesses are not counted in the clock. This latter ommission seems to stem from a desire to obtain a load-independent measure of processor usage. The primary effect of the memory access count will be to hide memory interference effects due to load; this could well

avoid a 10-20% variation between charges for the same program. However, the load-dependent part of page and segment- descriptor table accesses should amount to something less than 1% of the total acceses; on the other hand, ignoring page and descriptor table accesses completely could mean that 10-20% of the processor and memory controller usage would be unaccounted for, independent of load. It has been agreed, therefore, that the timer should count page and descriptor table accesses as well as other accesses.

If all the above changes are made as planned, the proposed accounting clock system is satisfactory and acceptable as an interim clock system, with the understanding that a redesigned memory controller is required to produce a truly satisfactory system.