1. "Clock Controller" does not look enough like a Memory Controller to convince me that the Clock could be put into a Memory Controller later.

2. Calendar Clock Interrupts cannot be masked. This is intolerable.

3. Calendar Clock Interrupts cannot be given priority between other interrupts without leaving arbitrary holes and screwing up the interface. This is intolerable.

4. In general, the "Clock Controller" loses up the clean modularity of the interrupt design.

5. There is too much emphasis on duplexing within the clock, not enough on distinct clocks. If the Clock Controller goes down, Multics is down. It seems to me that the Clock Controller is at least as complex in logic as the clock itself.

6. Fuchs apparently missed the point on independent power. We need power from commercial lines, but independent of MC not for the basic clock module only, to keep it ticking over if the system is shut down for a vacation. (or MC maintenance) We seem to think service through a power failure is the problem.

7. Having the Processor Interval Timer ignore segment/page cycles seems like we are buggering the meter too much. The numbers we get are useless, statistically, except for BTL billing.

8. Is all this reliability going to stack in trouble or an attempt at a snow job? It doesn't make much sense.

9. Would like to employ a clock model from MC1 and plug it in to MC2 if MC1 is down.

10. We want Calendar Clock accessible ~~only in~~ in Slave Mode. Switchable only in Master Mode, if at all.

11. Why only not upper 36 bits? Upper 36 allow setting from a tape (WWV, etc.) 1 min Vax.

12. Two ports is not enough. Need ability for more than 2 processes + GAC + OPCOM if they desire.

Clock controller needs 8 ports.