

From: W. Olin Sibert  
To: MTB Distribution  
Date: October 18, 1980  
Subject: Desupporting the Bulk Store,  
(or) Whither Page Multi-Level?

This MTB discusses the reasons for and consequences of eliminating support for the Bulk Store on Multics. The most pressing reason involves software development for the ORION, but there are other reasons as well. Please direct any comments or questions to the author:

By Multics mail at MIT or System-M to:

Sibert.Multics

Or by mail to:

W. Olin Sibert  
Cambridge Information Systems Laboratory  
HED MA22

HVN 8\*261-9353

---

Multics Project internal working documentation. Not to be reproduced or distributed outside the project without consent of the author or director, Multics System Development.

## Introduction:

In designing Page Control for the ORION, it has become evident that something must be done to compensate for the decreased number of "software" bits in the ORION PTW, as compared to the Level 68 PTW. There are several possibilities, the most attractive of which is to salvage the bits currently used to support the Bulk Store. This is not a major handicap, since there will not be a Bulk Store or similar device in the ORION configuration.

There has also recently been input from Marketing which suggests that it would be worth considering elimination of support for Bulk Store for MR9.0. This is hardly surprising; Bulk Store is used by an ever decreasing number of systems, and daily becomes harder to maintain.

Finally, as the need to support larger and larger disks comes upon us, it is necessary to change page control in a major way to support them. This would be considerably easier to do without having Bulk Store support in the way; additionally, removal of Bulk Store support would free up several PTW bits for use as larger disk addresses.

The most important reason, however, and the reason why the preceding suggestions should be considered, is that the economic justification for the existing Bulk Store has completely disappeared.

In response to these reasons for eliminating the Bulk Store, this MTB explores the software implications of decommitting Bulk Store support, and also examines the change from a hardware technology viewpoint. This MTB discusses a possible plan only; the actual decision of whether or not to eliminate Bulk Store support will be made by the Multics Marketing organization. Therefore, this MTB should not be considered to be a definitive plan, and should not be discussed except on that basis.

## Software Implications:

The present Bulk Store support exists for the convenience of an ever-decreasing number of systems; only a few systems remain today which have Bulk Store. It makes all of page control significantly more complicated to support Bulk Store; removing it would make the code smaller and simpler, as well as providing an efficiency improvement in page faults.

Elimination of Bulk Store support would make wired page control perhaps 30 percent smaller, with a consequent improvement in maintainability. The efficiency improvements for page faults might permit recovery of 5 percent of page fault time. Finally, a small amount of wired storage would be freed up (even for systems without Bulk Store). These estimates are very tentative, and may be quite inaccurate; however, the important facts are that the code will definitely become smaller and faster. Removing the Bulk Store can only make things better.

In order to support large disks in page control, or to support the ORION PTW format at all, some more bits must be available in the PTW for software use. One could be easily salvaged from the add\_type field today, but that's not enough. Removing the Bulk Store support salvages four more -- two from the add\_type field, ptw.nypd, and ptw.first.

Another possibility for recovering PTW bits is to move some of them to the Core Map Entry for the page; this would give us three or four more. At least one of these changes must be made, however, and they are both a considerable amount of work. There are simply not enough bits available for simple solutions to be applicable. I think the removal of Bulk Store support is the easier of the two.

It may be that removal of Bulk Store support does not provide enough PTW bits to support really large disks; in this case, it will also be necessary to move some bits to the CME, but this will be easier to accomplish once Bulk Store support is no longer in the way. In any case, however, enough bits will be freed up by the initial change that we will be able to support disks eight times larger than we can today.

Support of the Bulk Store is presently rather problematical, anyway: there is no longer any at CISL to test with, so it may be that we will make some other change which will break the support. Furthermore, and more importantly, we no longer have any exposure sites with Bulk Store; we are in the position of supporting a piece of special purpose hardware which we cannot even expose to a user load. If it proves necessary to avoid acute customer distress, it would be possible (though highly undesirable) to retain a version of page control which supports Bulk Store, derived from the present system. This version could be used as a special, to support existing Bulk Stores and still allow upgrades to later releases of non-supervisor software.

If this is done, some modules outside bound\_page\_control (in all systems, not just the specials) will retain code to support the Bulk Store which could otherwise have been deleted; this additional code would be slightly in the way, and would require some work to avoid, but not very much. Once Bulk Store is completely gone, the last vestiges (in initialization, mostly) could be eliminated easily.

Needless to say, this is not an attractive alternative. If we are to remove the support for the Bulk Store, it should happen, and be done with. We should not be in the business of having special supervisors for special customers.

### Advances in Hardware Technology and Bulk Core:

In light of present technology, existing page-multilevel is basically obsolete. It was designed to deal with economic considerations which no longer hold true as they did ten years ago. MOS main memory and disk memory technology are improving rapidly, and have now well surpassed bulk core. The only reason to continue support of Honeywell Bulk Store is that existing customers have it. No new product should have to support it, either in hardware or software. None of the new Honeywell hardware products do support it, so the software should be eliminated if possible.

Bulk Store as presently implemented is not a viable hardware technology -- MOS memories abound today which are many times larger than the largest Bulk Store in existence, and they are cheaper, easier to maintain, and smaller than bulk core. The Bulk Store is difficult to maintain, consumes much power, and takes up an inordinate amount of floor space.

It is likely that this situation will continue for some time -- there is much more demand for MOS memory technology than for marginally slower, marginally larger bulk memory. Even if some form of bulk memory (such as magnetic bubbles) becomes available and cost-effective, it would have to be several times larger than main memory in order to be cost-effective, at which point it becomes similar in both access time and capacity to ordinary disk drives.

Existing page-multilevel algorithms will not perform effectively when dealing with a 100 MW bulk memory. Furthermore, the data base organizations would break down completely, requiring a complete redesign of the data formats, since they would no longer fit in the same segment, as is done today.

Note: Honeywell Bulk Store is relatively fast to access (few microseconds), and performs multi-word transfers rapidly. It could be effectively used as "slow" main memory, and would be considerably simpler to manage if it were directly (SCU) addressable in this fashion. The REAL reason that it exists as an I/O device at all, rather than being slow main memory, is that machines of two decades ago simply couldn't address millions of words of main memory. Ah, progress. It seems likely that future "intermediate" bulk memory will not have the same fast random access characteristics of bulk core, but will be somewhat slower to access, but very fast to transfer.

### Future Possibilities for Multi-Level Paging Hardware:

This is not to say that the concept of multilevel paging is obsolete; it may be that future technology will provide other devices in which to implement it. This final section discusses the possibilities I see for future Multi-Level storage hierarchy hardware; it is not intended as a complete or thorough description of all the possibilities, but rather just as a brief overview of a few.

I think there are two major possibilities for future Multi-Level storage hardware: large (100 MW and up) bulk memories (magnetic bubble, for instance), and very large (trillion bit storage) storage devices, such as the Ampex Terabit Memory, the IBM 3850 "honeycomb" mass storage, or optical disks. Other possible technologies seem too undeveloped at this point to be worth consideration.

Bubble memory bulk storage, for reasons described above, seems unlikely in light of improving disk and MOS technology; if it existed, however, it would be desirable to manage it in some fashion other than page by page, due to the huge number of pages involved. The major advantage of magnetic bubble memory seems to be its non-volatility, and we can accomplish the same thing today with batteries and MOS.

Other possibilities for intermediate bulk memory are more far-fetched -- IC technology has left Charge Coupled Devices far behind, as they have proved difficult to produce and not as fast as originally hoped. Electron Beam storage, a far cry from the Williams Tube memories of the Fifties, is still very experimental; while it offers great densities and bandwidth, it also doesn't exist commercially, and there does not seem to be a great likelihood that it will anytime soon.

The mass storage possibility is much more likely, and is much more desirable as well. Again, however, it would certainly not be managed on a single page basis. Possibly the right level at which to support a mass store is at the logical volume level, but it requires further study. Some sort of mass storage hardware would make a very attractive addition to Multics, and should certainly be considered.

Large mass storage could take the form of read/write storage, such as the Ampex TBM or IBM 3850, or might be simply large scale read-only, or write-once, read-later optical memory. The read/write mass storage devices are largely based on some form of automatically handled magnetic recording technology -- they have access times measured in seconds, but are very fast to transfer (>100 Mbyte/second).

Optical technology has the potential for much faster access (tens or hundreds of milliseconds), and also has very high potential bandwidth. Its major disadvantage is its read-only, or write-once, nature. It would be ideal for accessing large volumes of constant information, and, in the latter case, for doing backup. Also, of course, when there's "enough" storage, it doesn't matter that much that it can only be written once, since it will "never" run out.

**Conclusions:**

I think the Bulk Store should be eliminated from Multics configurations everywhere. The benefits in software are considerable, and the hardware itself is completely obsolete. Support of large disks is much more important than Bulk Store support. None of the future possibilities for bulk memory seem to fit the same model as Bulk Store did, and therefore the Bulk Store management algorithms are not applicable to them, and can be discarded.