XEROX
BUSINESS SYSTEMS
System Development Department

To: Metcalfe, Belleville, Garner

Date: September 25, 1978

From: W. C. Lynch

Location: Palo Alto

Subject: Addresses in IOCBs

Organization: SDD/SD/SSW/Pilot

Archive document #: Binkley archive number

Keywords: OIS PrinceOps, IOCB, Wildflower

Filed on: [Iris] < Lynch > IOCBPrincOps.Memo

Copies: Archives, Irby, Kennedy, Lampson, Liddle, Lynch, Thacker, Townsend, Wick

As a result of a recent conversation with Butler Lampson on the topic of I/O on Wildflower, a number of issues concerning the extent to which Wildflower meets the current OIS PrinceOps arose. The tone of the issues is that the current PrinceOps may not be suitable for the more modest processors in the OIS line. These issues all have to do with the question of whether or not the buffer addresses in CSBs and IOCBs should be virtual or real.

As I see it the issues are:

1) If the buffer addresses are virtual, when are they bound to real addresses? The current PrinceOps is silent on this point and might lead one to believe that it is possible to race map changes against the I/O itself. The current implementation of the D0 supports this to a large extent by means of its mapping hardware. I believe that this is excessive and useless generality and that virtual addresses should be considered bound to real address no later than the fetch of the IOCB for processing. Any map changes between that point in time and the completion of the IOCB would lead to undefined operation. I could easily be persuaded that the map should not be changed under an IOCB while it was chained on the CSB for processing.

2) If the buffer addresses are virtual and are bound (translated) at the fetch of the IOCB, what happens when the buffer crosses a page boundry?

There seem to be three possibilities:

a) The processor is fast enough or powerful enough to rebind the incremented virtual address at each page crossing. (The D0 uses a more extreme measure in that the memory control is powerful to rebind at each (I/O) memory reference.) This may be too severe a constraint for small OIS processors.

b) Buffers are not permitted to cross page boundries. (We are now coming to the view that buffers should not cross megabit boundries on the D0). On the Wildflower this seems acceptable for all but the display bitmap, which would have to be handled by one of the other techniques. On the D0, the display bitmap and perhaps IIT and IOT are exceptions. In any case Pilot must be modified so that its space allocation interface supports the allocation of spaces having suitable properties.

c) Real memory is allocated to the buffer so that the virtual memory of the buffer has contiguous real memory allocated underneath it. This has serious negative connotations for the real memory allocator in Pilot, which must now become much smarter on the smaller machines. It also resurrects the spectre of *sandbars* which we tried to leave behind in going to virtual memory in the first place.

3) If the buffer addresses are real, how are they bound and what happens when the buffer crosses a page boundry? The ideal place to perform this translation would seem to be in the routine that *pins* the buffer into real memory so that page faults will not occur. There are again three analagous cases:

a) The IOCB is transformed into $n+1$ data chained IOCBs where n is the number of page crossings. This is the scheme used on the System 360 and System 370. We know from those systems that there are some severe problems. First, the data chaining mechanism must typically be very fast. The sited systems ran into difficulties when new I/O devices and controllers were added *post facto* to the systems and proved too fast for the existing data chaining mechanism. It is not clear that the data chaining can be made significantly faster than retranslating the virtual address, case 2a) above. Second, an IOCB blows up into several IOCBs, necessatating the allocation of spaces in which they will be placed. Interactions between the translation process and shortages in this allocation have created serious problems. Third, the time consumed in software translation of the IOCB chain can be grossly excessive without very careful design.

b) Buffers may not cross page boundries (discussed above).

c) Contiguous real memory is bound under the buffer (discussed above).

There is nothing to prevent us from adopting a mixed strategy other than our ability to be clear about stating that strategy. In any case we should come to some strategy and have it incorporated into the PrinceOps before it is released as the basis of an LSI design.