Title
PHILOSOPHICAL AND PRACTICAL STANDARDIZATION OF COMPUTER INPUT/OUTPUT CONNECTIONS

Author
Meng, John D.

Publication Date
1969-10-01

Submitted to IEEE Transactions
Nuclear Science

UCRL-19378
Preprint
cy. 2

PHILOSOPHICAL AND PRACTICAL STANDARDIZATION OF COMPUTER INPUT/OUTPUT CONNECTIONS

John D. Meng

October 1969

AEC Contract No. W-7405-eng-48

LAWRENCE RADIATION LABORATORY
UNIVERSITY of CALIFORNIA BERKELEY

PHILOSOPHICAL AND PRACTICAL STANDARDIZATION OF COMPUTER INPUT/OUTPUT CONNECTIONS

John D. Meng

Lawrence Radiation Laboratory
University of California
Berkeley, California

SUMMARY

Philosophically, small computer input/output connections can be made to a standard (such as CAMAC), rather than directly to a particular computer. Where manufacturer-produced software is either non-existent or can be sacrificed without heavy losses, a bigger computer may be practical with money saved from in-house interfacing. Such in-house peripheral interfacing can be made even more economical by practicing interface standardization. An example is discussed with solutions to the major problems involved.

INTRODUCTION

Nuclear Instrumentation Modules have become an international standard for many types of hardware. Computer interfaces have escaped standardization to a large extent, however, because of the many variables which must be considered. These include word length, memory size, different types of input/output routes and different logic levels, as well as many others.

Determinations for these variables are often made at the whim of the computer manufacturer. Today, however, computer utilization seems to be reaching a stage of sophistication which forces users to request larger and more powerful central processors.

If computer interfaces to peripheral devices were to be standardized, direct interchangeability of peripherals might become feasible among many computers. This would conceivably ease the financial burden of upgrading systems, since a complete set of new peripherals (with a complete set of new interface designs) would not then be required when switching to the newer and bigger computer (see Fig. 1).

Presently being in the midst of just such an upgrading, we have carefully considered (mostly in hindsight but partially in contemporary perspective) what such standardization would have meant to us.

BACKGROUND

A central processor communicates with two types of peripheral. The first of these is, for all practical purposes, a domestic part of the computer - the computer talks to it, listens to it and relies on it for support. Such peripherals commonly include magnetic tapes and discs, as well as some more exotic but less common devices. Magnetic tape is particularly useful because it is a large chunk of computer memory which the operator can easily remove and carry back to his office for safe-keeping.

Second is the foreign device - a pulse-height analyzer, scaler, ADC, keyboard, display or some other real-time data source or sink. Interface standardization seems to have been conceived with these devices in mind. They are often found in bunches at specific remote points; and often, though not always, produce data at unprodigious rates. Consequently, it may be practical to multiplex them together at their remote point and use only one link to the computer for each group of devices (Fig. 2). Standardization is uniquely useful where it is undesirable to devote these remote devices perpetually to a single, particular computer. Pulse-height analyzers, scalers and ADC's, for example, are often more generally useful if they can be plugged interchangeably into different types of computers or controllers. CAMAC, in fact, has been developed to facilitate this not only on an intra-laboratory scale, but on an inter-laboratory scale.

Thus, device controllers designed at Berkeley or Harwell might find use at Brookhaven or CERN, as well as the converse.

A question arises pertaining to the "natural" limits of such standardization. Specifically, should the interfaces to magnetic tape units or to disc memories be standardized as well as the interfaces to foreign devices. The answer depends entirely on who is building the interface to the disc or to the magnetic tape unit. There are certainly many discs and tape units on the market which are non computer-specific. That is, they are not sold to be connected to one, and only one, type of computer. Rather they are manufactured and sold to be connected to as many different models of computer as possible.

The interface is the computer-specific device. When you change computers, the interface is the unit which must be changed to connect peripherals to the new computer. If the user is faced with the interfacing chore, he wants to simplify, as much as possible, conversion to a new computer. If he builds his peripheral interfaces (independent of any computer) to connect to a standard (such as CAMAC), and then builds a computer-to-CAMAC connection for any computer he wants to connect to these peripherals, he will minimize his redesign effort (Fig. 1). Connecting a new machine to an old peripheral becomes a job of designing and debugging one computer-to-CAMAC connection. Additional old peripherals can be connected by duplicating this single design.

Consider for example, buying computer A, and interfacing a disc manufactured by Absorbadisc Corporation and a tape machine manufactured by Tapit Corporation. Then interface a few CAMAC bins to the computer to take care of scalers, ADC's and similar devices. It is now sensible when designing the Tape and Disc interfaces to design them to mate to the CAMAC standard instead of to the computer directly. The reason is that when computer A is retired or possibly sold into slavery to some bigger and faster machine (computer B), computer B may have use for computer A's peripherals. Since computer B will undoubtedly have a CAMAC interface designed onto it to care for foreign peripherals, the domestic peripherals can be connected to computer B with an absolute minimum of hardware redesign or debug.

In fact, if bondage to computer B is the destiny of computer A, this bondage becomes a fairly simple matter of a CAMAC-to-CAMAC interface after both machines are outfitted with CAMAC hardware. Indeed, if programmers become accustomed to programming to CAMAC bins instead of to peculiarly individual devices, even the programming transition might be simplified.

If the computer manufacturer is building interfaces, his profit is maximized in many ways by having his particular interface adopted as a "standard". Among other things, this tends to force users to buy new machines not so much on the criteria of their hardware virtues as by inertia. By adopting company A's interface as an in-house standard, company B's products can be resisted with the argument that the in-house standard will become obsolete. Competition based on product virtue rather than on such inertia is generally beneficial to the user in the long run, and a user-dictated standard interface (such as CAMAC) encourages this.

A prominent exception to the use of the standard interface on domestic peripherals does exist. If a computer is sold as a computer-peripheral-software package whose usefulness will be drastically impaired or destroyed by breaking up the set (by not buying the peripherals for example), it may be less costly in the long run to buy the entire package, saving programming expense but spending more for hardware.

I hope, at this point, to have triggered some familiar chords in your repertior of problems. I will devote the reaminder of this discussion to our various experiences at LRL, illuminating our many mistakes with precious rays of hindsight.

Those of you who are familiar with CAMAC have undoubtedly become uneasy. I have, to a certain extent glossed over some of the hardware problems (many peculiar to any standard interfacing scheme) which arise very quickly when you begin to get your hands dirty in hardware design. Allow me to prematurely assure you, however, that we have encountered most of these in our current designs, and have in fact solved them.

Before going farther, perhaps a one paragraph description of CAMAC is in order.[1] CAMAC consists of a standard bin, similar in many mechanical respects to the NIM bin. In fact NIM modules will slide into the CAMAC bin (although the converse is not ture). The connectors at the rear of the bin (24 of them) are all wired in a standardized fashion. The twenty-fifth connector is not identical to the other twenty-four, and is reserved for a bin controller. The 24 connectors each have 24 pins reserved for data coming from the modules, 24 pins reserved for a data going into the modules, four pins reserved for selecting a part of a module and five pins reserved for specifying some operation to be performed (load a register, unload a register, clear a register, increment a register, etc.).

In addition to these bussed lines, each bin position is given two individual lines: one for signalling the bin controller when the module wants service, the other for signalling the individual module from the controller. Two bussed lines are used for timing pulses. Another buss wire is used to tell all the modules that the bin's data paths are busy, and finally, a buss is provided as a common response line from modules (useful, for example, where a single interrupt is all that is available to the system). The CAMAC bin contains a controller which is computer specific; that is, the controller is non-standard, and interfaces the computer to the bin.

Briefly, the substance of CAMAC is a bin containing 24-bit bi-directional information - transmission characteristics, all the necessary lines for multiplexing (busy, request and response lines) and a bi-directional command-transmission buss (fig. 3). These characteristics are sufficient and convenient for interfacing a peripheral or group of peripherals, foreign or domestic.

Hardware Dilemma

Getting down to specific cases, hardware difficulties do arise.

1. Different computers often differ in word length, among other things. How should a disc or tape machine controller be made flexible enough to accomodate two or more different word lengths?

2. We have been discussing a CAMAC bin containing 25 slots. In most instances, it is absurb to consider a disc, or even a magnetic tape unit, having to share a data input-output buss with potentially 8 or 10 other devices, when this buss may in turn have to share the computer with even more devices. The time-lapse between a service request and a computer answer can get intolerably long.

3. Assuming our devices may wish to utilize either a direct memory connection to the computer or a CPU-controlled connection or both, how can the module communicate this to the bin controller?

We have at this point been through the controller design problem twice. Our first attempts worked, and are operating a display-editing console and a card reader. The design was abbreviated in these two cases, however, because of concept uncertainties and the limited requirements of these devices. Our second attempt is a full fledged bin controller whose first test is to operate a disc controller. I call this a full-fledged attempt because the controller controls DMA as well as CPU data into and out of the bin. The computer is capable of manipulating the bin in a very general and flexible sense.

Now to specifics (Fig. 4). The word-length problem (1) was solved by breaking the controller (the disc controller in this case) into two modules interconnected by a non-standard one-to-one jumper cable on the rear of the modules. One of the two modules is disc-specific. That is, it is built to handle things which relate to the disc, independent of any computer involved.

The second module is a register module, containing a word counter, address register and a data register all the components related, more or less, to computer word size and core size. Data is serially fed to this module from the disc-controller module (one-bit characters) and the computer-sized word-assembly is performed here. When changing data word size, only this register module must be changed, possibly by moving jumper wires or changing internal switch settings. I will readily concede that word count and address registers are not always computer specific. By making them large enough, nearly any computer can be serviced. However, by having them in a separate register module, we expand our economics. The tape machine, for example, can use this same module in its interface (with very minor changes), as well as any other device communicating with the computer memory.

Problem (2) is the wasted-bin problem. If we buy a bin with back-panel wiring for 24 slots, and then restrict the bin to one device (for example the disc) we have a substantial waste. The solution is an abbreviated bin. Only wire enough slots to handle the bin controller and your device controller. The key to interchangeability is not the complete bin, but is instead the standard connectors and signals on the back side.

Our last problem (3) was bin flexibility. CAMAC specifies that half the function codes on the function lines be standard. These standards are generally oriented towards an order coming from the bin controller to the device controller. We have assigned some of the non-standard function codes to communication from the device controller to the bin controller. If a device requests service with the proper code on the function buss, the controller may interpret this function code to mean (depending on the specific code) possibly a DMA read or write is being requested, or an interrupt is being requested. Thereby the bin slot can request specific things of the bin controller, and the communication becomes, desirably, two way.

CONCLUSION

The above were our major conceptual problems in effecting our system, and our solutions to them. We have concluded that CAMAC is workable and valuable in many cases. At the same time, we see that it is not a general solution for every interfacing problem. Unusually high data rates and uncommonly long word lengths instantly defeat the system. But for most dealings with small computers, these are not the rule but the exception. CAMAC does promise to relieve many mundane, routine and boring chores (redesigns, re-redesigns and re-re-redesigns of peripheral interfaces), and offers unique economics to budget-conscious laboratories.

ACKNOWLEDGEMENT

I wish to acknowledge the help of John Kopf in overcoming many inertial and conceptual problems in the initial designs. Don Landis, Ed Lampo and Fred Goulding have been of particular help in developing effective communications with people not intimate with the problems. And Vicky Donelson must not be forgotten for her adroit prefiguration and exhuberant positive prepossessions.

FOOTNOTE AND REFERENCE

This work is part of the program of the Nuclear Chemistry Division of the Lawrence Radiation Laboratory, Berkeley, and was supported by AEC Contract W-7405-eng-48.

1. I. N. Hooton, and R. C. M. Barnes, A standardized data highway for on-line computer applications, in Proceedings of the Fall Joint Computer Conference, 1968, pp. 1077-1087.

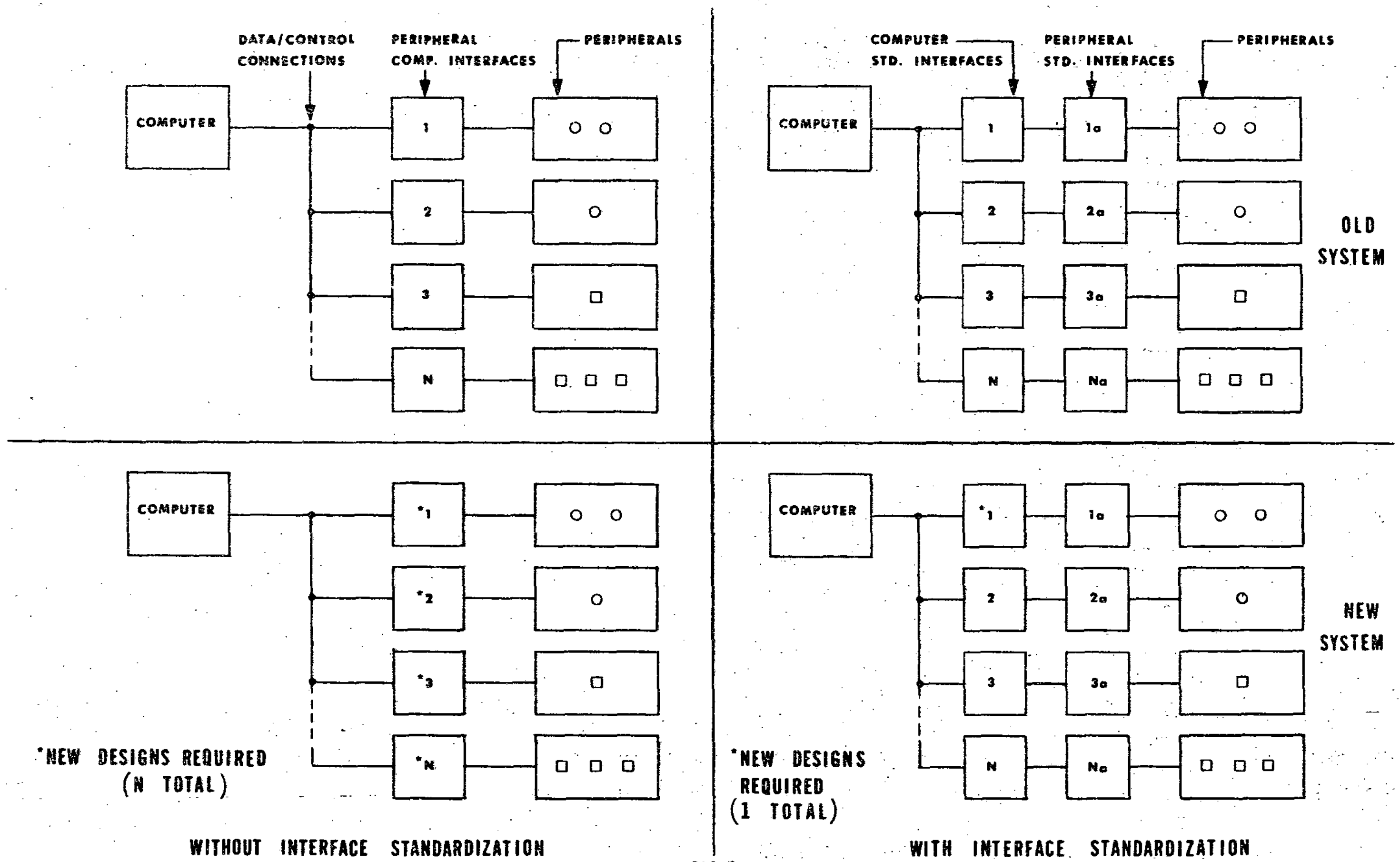

FIG. 1
THE ECONOMY OF STANDARDIZATION

XBL 6910-5787

Fig. 1

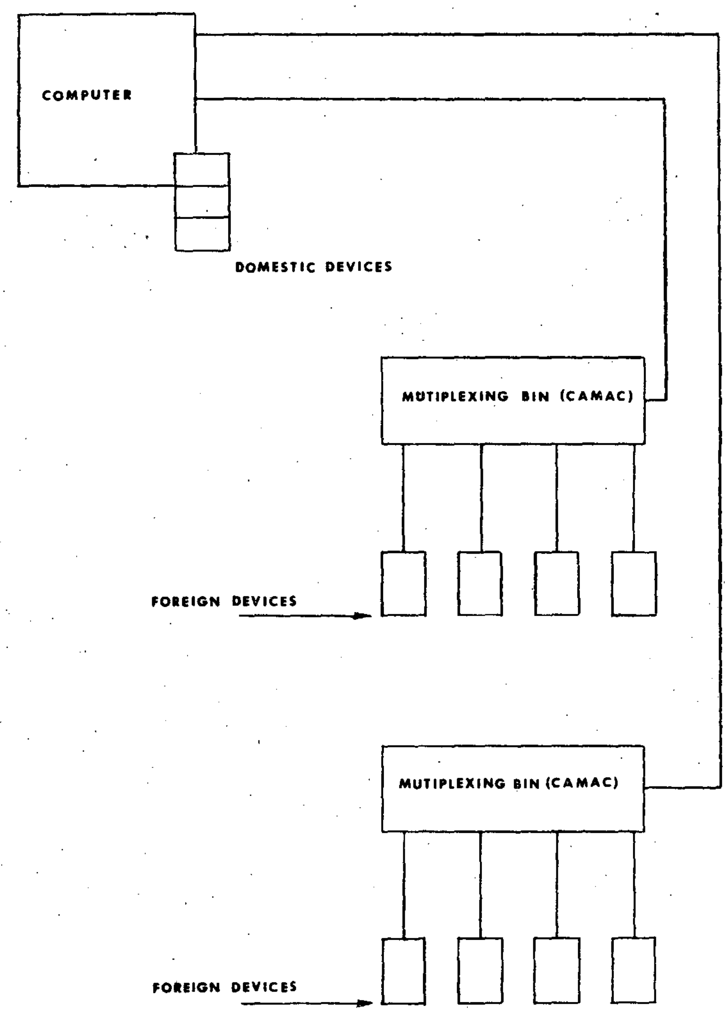

FIG. 2
FOREIGN AND DOMESTIC DEVICES

XBL 6910-5788

Fig. 2

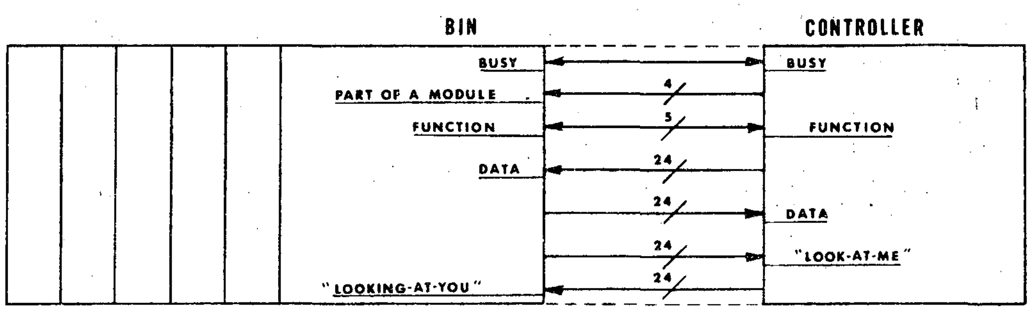

FIG. 3
SOME INFORMATION PATHS
IN THE CAMAC CRATE (BIN)

XBL 6910-5789

Fig. 3

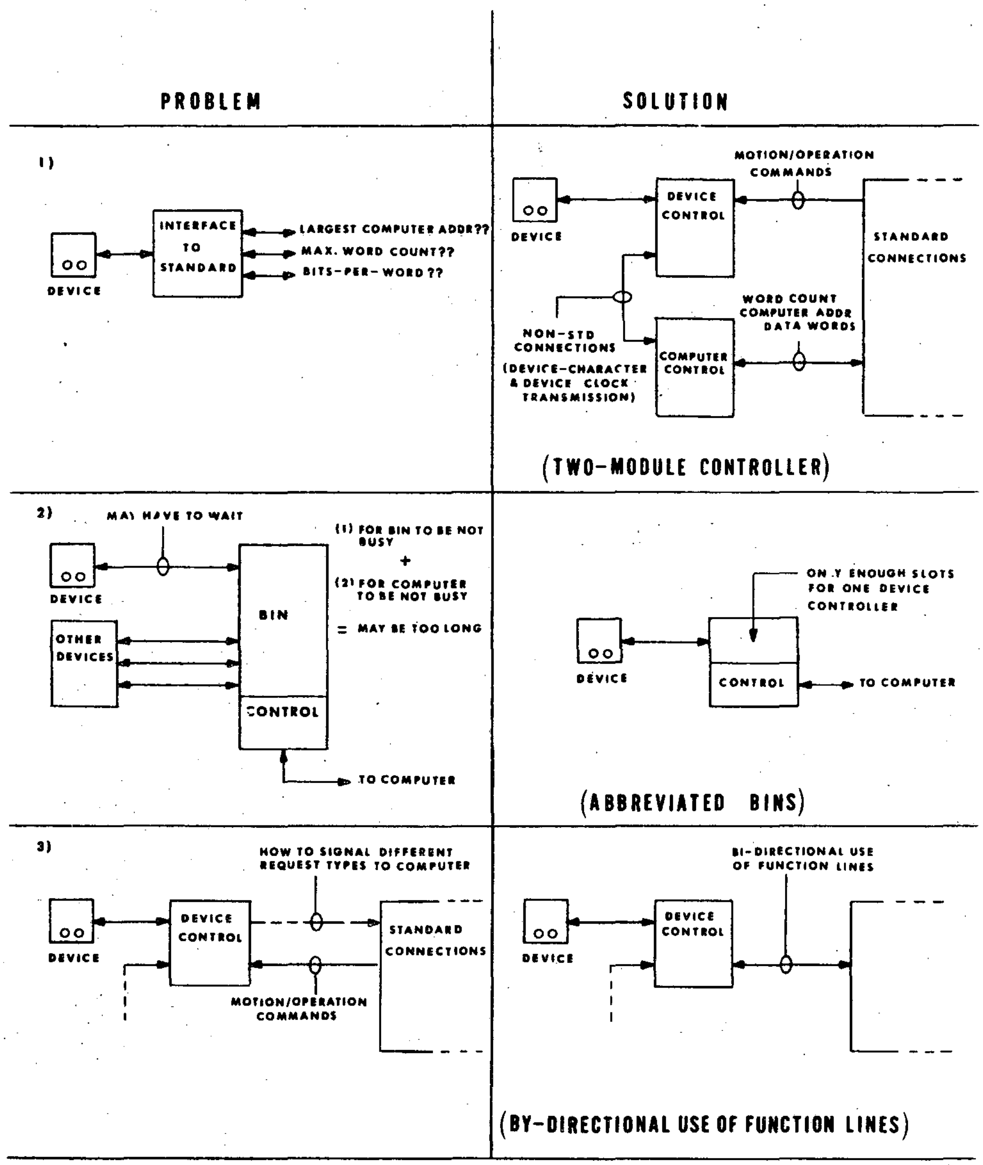

FIG. 4
PROBLEMS AND SOLUTIONS

XBL 6910-5790

Fig. 4

TECHNICAL INFORMATION DIVISION
LAWRENCE RADIATION LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720