June 1987

MTR10028
JSC 22603

Jerry K. Richardson

MIPS: The Good, The Bad, and The Useful

CONTRACT SPONSOR NASA
CONTRACT NO. F19628-86-C-0001 T-8114M
PROJECT NO. 852E
DEPT D115

The MITRE Corporation
Houston, Texas

Department Approval: W.P. Kincy Jr

W. P. Kincy, Jr.

MITRE Project Approval: Edwin S. Hern

Edwin S. Herndon

ABSTRACT

Many authors are critical of the use of (MIPS) Million of Instructions per Second as a measure of computer power. Some authors say that MIPS are meaningless. While there is justification for some of the criticism of MIPS, sometimes the criticism is carried too far. MIPS can be a useful number for planning and estimating purposes when used in a homogenous computer environment.

Comparisons between published MIPS ratings and benchmark results reveal that there does exist a high positive correlation between MIPS and tested performance, given a homogenous computer environment.

MIPS should be understood so as not to be misused. It is not correct that the use of MIPS is always inappropriate or inaccurate.

ACKNOWLEDGEMENTS

I wish to express my appreciation to several individuals who provided help and information for this document.

Bill Kincy and Stuart Bell provided much needed document review and publishing suggestions. Walter Bays and Danny Labasse provided articles and references related to MIPS and benchmarks. Jeff Lorentz provided editorial help and hints on how to structure various sections of the document.

Don Simanton (NASA sponsor) furnished the initiative for the selection of the topic, and also suggested the publication of this paper.

Special thanks goes to Jerry Trust who functioned as my principal sounding-board, technical advisor, and helpful critic. Without his help, this document would not have been completed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ACRONYMS

CISC	Complex Instruction Set Computer
CP	Control Program
CPU	Central Processing Unit
DASD	Direct Access Storage Device
DP	Data Processing
DPSD	Data Processing Systems Division
ETR	External Throughput Rate
ITR	Internal Throughput Rate
JSC	Johnson Space Center
MFLOPS	Millions of Floating Point Operations per Second
MIPS	Millions of Instructions per Second
MP	Multi-processor
MTR	MITRE Technical Report
NASA	National Aeronautics and Space Administration
PROFS	Professional Office Systems
RISC	Reduced Instruction Set Computer
RPMs	Revolutions per Minute

This page left blank intentionally.

SECTION 1

INTRODUCTION

This paper was written in response to a request (May 1987) from our NASA sponsor that MITRE prepare a paper discussing quantitative analysis of MIPS, along with any usefulness, or non-usefulness, of MIPS in comparing large and small computers, and computers with like and unlike architectures. The initial response was simply the kernel of this paper and was written as a four page PROFS note and mailed to the various interested individuals.

DPSD Division Chief, Don Simanton, subsequently suggested that I "clean-up" the PROFS note and make it into a publishable paper. This MTR is the result of that suggestion

1.1 GENERAL

Many articles and presentations have pointed out the inappropriateness of using MIPS (Millons of Instructions per Second) as a measure of performance or power of a modern large scale computer system. The following is typical of many such discussions

> Performance rating implies a measure. Traditionally, the performance measures used by the industry have been either cycle times or instruction execution rates (in millions of instructions per second MIPS), the instructions here being machine level instructions. These measures today are not only misleading, but downright irrelevant as the measures of the performance power of today's data processing systems. [1]

Very few articles are objective and balanced in their discussion of the utility of MIPS as well as the dangers and difficulties with the use of MIPS. Many authors make stereotyped and senseless statements concerning MIPS and then proceed to use them extensively , and with good effect.

> MIPS (meaningless indicator of processor speed) is, of course, not a good measure of relative CPU power, but for our purposes, it will work fine, since we are only interested in relative performance and not actual numbers. [2]

Most discussions on MIPS fail to point out, or adequately treat the following:

1) The inevitable, wide variations in MIPS across non-homogenous architectures or non-homogenous workloads does not mean that MIPS cannot be used with relative confidence in homogenous computer environments.

2) There are at least two kinds of useful MIPS available. The most common notion for MIPS is in the context of an aggregate or composite rate of instruction execution. But, one may also be interested in MIPS numbers (instruction execution rates) for individual machine level functions (load/add, move, etc.).

3) MIPS does not have to be a measure of system power (total throughput capacity) in order to be a useful estimator of relative CPU power.

4) Obtaining performance estimates that are superior to MIPS is often expensive and time consuming. And the difference in accuracy may not justify the cost.

1.2 PURPOSE

The primary purpose of this paper is to demonstrate that MIPS is a good indicator of relative CPU performance in a homogenous computing environment. Homogenous means similar architectures and equivalent workloads.

This paper will assist the reader's suspicions of an aggregate MIPS number as a measure of CPU power across an undefined range of computers. At the same time this paper will encourage the reader to realize that there is an equally legitimate use of aggregate MIPS numbers when they span a range of homogenous computers.

This paper will discuss both single-function MIPS numbers and aggregate MIPS numbers. Normally a weighting process is used to form an aggregate MIPS number; or weighting occurs due to the statistical composition of the workload if MIPS is measured rather than calculated. In other words, the rates for individual instructions such as loads and adds are combined in a weighted average fashion to form an aggregate MIPS number.

The terms 'aggregate MIPS' will be used synonymously with 'MIPS' and both will refer to an aggregate instruction rate comprised of different single-function instruction rates. The term 'single-function' MIPS will refer to an instruction rate for an individual machine level function such as a load or an add.

1.3 SCOPE

This paper will be concerned with MIPS as it pertains to digital, scalar computers. For the comparison of vector or array processors we would use Millions of Floating Point Operations per Second (MFLOPS).

Although a discussion of MFLOPS has considerable similarity to a discussion of MIPS; MFLOPS as a measure has not yet drawn the widespread opprobrium that MIPS has drawn even though it is rather obvious, upon examination, that MFLOPS alone is not sufficient to compare the performance of two vector processing computers of dissimilar architecture or with dissimilar workloads.

Note that some authors classify both MIPS and MFLOPS as useless numbers.

> Measuring the performance of a machine is a complex issue. The inaccuracy and danger of using a simplistic measure like MIPS is widely known. Unfortunately it is still used, even though more accurate measures such as Internal Throughput Rate (ITR) are available. In the world of vector processing, the situation is no better.
>
> The most common measurement used to compare vector processing speeds is MFLOPS, pronounced "MegaFLOPS," which means "millions of floating-point operations per second." Many modern vector machines can also do integer and logical operations on vectors, but it is the floating-point operations that consume the most time in scalar mode, and they are used to measure performance. This is still true, although even computer salesmen recognize that "to quote MIPS is dangerous; to quote MFLOPS is suicide. [3]

There are many scholars who would not agree with the type of blanket, unqualified criticisms offered above. R.W. Hockney and C.R. Jesshope in their excellent book, PARALLEL COMPUTERS [4], use MFLOPS as one of two parameters for comparing computer performance.

> The two parameters [the half-performance length and the maximum or asymptotic performance] ... completely describe the hardware performance of the idealized generic computer and give a first-order description of any real computer. [5]
>
> The two parameters...provide us with a quantitative means of comparing the parallelism and maximum performance of all computers. [6]

The operative phrases in the quotes above, for our discussion, are "first-order description of any real computer" and "quantitative means of comparing". This is the reason for wanting to use a number like MFLOPS or MIPS in the first place.

It is instructive to note that Hockney and Jesshope imply at least three parameters in comparing the performance of vector processors. One of the parameters [the half-performance length] is a measure of parallelism in a computer; the second parameter [maximum or asymptotic performance] measures the theoretical maximum computational speed in MFLOPS. The third (implied) parameter is workload

> for both the CYBER 205 and the CRAY1. The comparison of these machines is difficult because their performance is strongly dependent on the problem being solved and the manner in which it is organized and programmed. [7]

Without assumptions concerning parallelism and workload, MFLOPS alone is not sufficient to compare the power of two computers However, if one does know or make assumptions concerning the amount of parallelism and the workload, then MFLOPS is a reasonable number to use to compare computer power. Similar assumptions are needed when MIPS is used to compare computer power.

The recurring themes that are always interwoven in any realistic comparison of computer performance are 1) speed, 2) architecture, and 3) workload characteristics. Whenever one of these themes is selected as a basis of comparison, the stated or underlying assumption is that the other two are comparable.

If the reader is interested in relationships between MIPS and MFLOPS for a given computer running a given workload. Literature on the DENELCOR HEP supercomputer [8] shows some tested relationships between MIPS and MFLOPS on the DENELCOR HEP.

SECTION 2

MIPS, INTERNAL THROUGHPUT RATE, AND POWER

2.1 Use of MIPS

The desire by planners to use MIPS as a single, simple measure of computer performance is as reasonable as the desire to know the maximum possible speed of a racing car, or perhaps the rated horsepower of a racing car's engine. In that sense, it is not unreasonable to expect that one could derive or measure a single number that would express the power delivery capability of a machine, even if that machine happens to be as complex as a modern digital computer

On the other hand, to expect that a single number could capture how well a racing car would start, steer, and stop or how well it would fare in the Indianapolis 500, is simply expecting too much from a single measurement. The same is true for computers.

There is no perfect, single measure for the performance of a complex digital computer However, single number estimates of power are needed and used Capacity planners, computer performance analysts, system modelers, and those concerned with procuring new systems or upgrades to existing systems need (and will find, guess, or manufacture) planning numbers. These numbers certainly can be, and usually are, ballpark estimates with a precision sufficiently appropriate for planning and modeling.

The Wadsworth's article [2], referenced earlier, is an example of a well written article that refers to MIPS more than a dozen times in the course of developing a practical methodology that "uses readily available numbers to determine the effect of adding various pieces of hardware to the DP center to reduce on-line response times". And what was one of the most important of those readily available numbers used? MIPS!

Still, before Mr. Wadsworth would use MIPS, he made the disclaimer quoted earlier. How can MIPS be "not a good measure of relative CPU power, but for our purposes, it will work fine, since we are only interested in relative performance..." ? In other words, MIPS is not a good measure, but it will work fine. That's a reasonable statement of the argument this paper is making.

2 2 INTERNAL THROUGHPUT RATE

Internal Throughput Rate (ITR), which can be interpreted as a measure of CPU power (for a given workload) is the sort of performance measure that we would like to be able to estimate with a MIPS number. Although they are not perfect, benchmark results are preferable to MIPS. Whenever possible, users should conduct their own benchmark runs on target computers in order to insure that the test environments map the proposed real environments . Benchmarking is, however, time consuming and expensive; also some users may not have access to various vendor's test facilities.

A timed test represents the completion of some sort of actual end user defined work; hence, a meaningful measure of power can be derived from such a test. IBM, along with numerous other vendors and CPU testers, uses the concept of Internal Throughput Rate which is defined as the number of completed jobs or transactions (commands) per processor busy second. The count of completed transactions in unit elapsed time (wall clock time) is usually referred to as the EXTERNAL THROUGHPUT RATE (see EQUATION 1).

EQUATION 1:

$$\text{INTERNAL THROUGHPUT RATE (ITR)} = \frac{\text{(EXTERNAL THROUGHPUT RATE)*(100)}}{\text{(\% TOTAL SYSTEM (CPU) BUSY)}}$$

To determine ITR, throughput is measured in a context which prevents I/O or memory configurations from being a constraint If non-constrained measurement conditions are not provided, then the time that the CPU spends waiting on I/O or memory is factored out by the use of the above ITR formula.

ITR can be very useful and accurate as a measure of CPU power when comparing computers with similar workloads and measurement conditions. As with any other measure however, the misuse of ITR may result in faulty conclusions.

> Internal throughput is a theoretical measure of processor capacity. An assumption is made that if the processor were driven to 100% busy in an unconstrained environment, the projected ITR numbers could [be] delivered as ETR [external throughput rate]....
>
> Here we see a situation in which neither MP system reached the throughput indicated by the ITR numbers. There are three factors causing this:
>
> o DASD Contention
>
> o Increasing CP busy time per transaction
>
> o Main Storage Constraint....
>
> Here we see a case where we were unable to achieve the ITR rates when the processor was driven to capacity. The measured ITR of the 5880 processor, even at 74% utilization, still projected a capacity about 11% greater than the maximum possible There was no way to accurately predict the size of this effect except by measuring it. The point here is that if we had planned our capacity based on ITR projections, we would be out of capacity far sooner than expected. [9]

ITR may be invalid unless the CPU is actually tested at or near 100 percent busy in order to see whether or not any unsuspected constraints exist. If ITR is calculated with the CPU busy considerably less than 100 percent, there is no guarantee that the theoretical ITR can, in fact, be accomplished. The major problem, of course, is that in many cases it is difficult to construct a benchmark and secure the necessary DASD and memory to cause a given processor to stay close to 100 percent busy during the entire run (excluding tight looped routines that consume only CPU resources).

2.3 PROBLEMS WITH THE USE OF MIPS

2.3.1 The Problem of Aggregate Measurements

One of the most, if not the most, serious problems with MIPS is the fact that it is supposed to be an aggregate number. This problem of MIPS being an aggregate, single number descriptor is perhaps not obvious enough. ITR, though a better measure, also suffers from being an aggregate, single number descriptor of computer performance.

The best way to describe the performance of a computer is with a vector of numbers, each of which measures a different aspect of computer performance. However, a descriptor vector must be eventually converted into a scalar (single number) if one is interested in making an overall evaluation or making a competitive comparison among two or more machines. An aggregate MIPS number can vary depending upon the instruction mix used to generate the number, or (what amounts to the same thing) an aggregate MIPS number can vary depending upon the weight assigned to the individual instructions that comprise the MIPS number.

The issue of rational aggregation is non-trivial. Suppose one wishes to perform an overall comparison (ranking) among three or more processors based upon their execution times for a multiple set of machine level instructions; it turns out that there are some common perceptions of rationality that cannot simultaneously be preserved. Nobel Laureate, Dr. Kenneth Arrow formally proved this proposition in what is now called Arrow's Theorem (also called Arrow's Impossibility Theorem) [10]. A slightly modified version of Arrow's rationality perceptions stated in computer processor terms are the following:

1) Any number of processors is allowed and any number of machine level functions may be specified in a composite evaluation scheme designed to produce an overall ranking.

2) If processor A ranks higher than processor B in every single machine level function used in the evaluation, then processor A must rank higher than processor B in overall ranking.

3) There shall be no dictatorial function in which processor A outranks processor B while A is being outranked by B in all other functions. That guarantees that processor A will outrank processor B in the overall ranking

The somewhat shocking results of Arrow's Theorem is that there is NO SCORING SYSTEM POSSIBLE that will preserve the above stated rationality perceptions. That is, if you wish all of the above rationality perceptions to function jointly as rationality requirements in an evaluation scheme, you are out of luck. It can't be done. Which does one give up in a ranking or scoring scheme if MIPS are being used as an aggregate measure? Rationality perception number three is the one that is sacrificed with the concept of aggregate MIPS.

In other words, it is unavoidable that an aggregate MIPS number MAY be so influenced by one particular machine function (because of the instruction mix or assigned weights) that it will not matter how all the competing processors actually performed (ranked) on all other functions used in the evaluation. Also, it is conceivable that a given machine's architecture could be so constructed that a particular processor would make such a tremendous score in one machine level function that it would not matter that it ranked dead last in all other functional categories.

Requirement three cannot be preserved without giving up one of the other rationality requirements. Which one do you give up? There are no good choices from the standpoint of equitable evaluation.

We have to live with the fact that an aggregate MIPS score can be unacceptably driven by one of the functional parts that make up the aggregate. This is potentially one of the most serious problems with the use of aggregate MIPS numbers. Within the bounds where MIPS is useful, it will seldom be a problem. However, it is one of the reasons why it would be helpful if vendors would publish aggregate MIPS numbers for their processors along with the single-function MIPS numbers (or times) that make up the aggregates. But most vendors have no interest in doing this; they are more interested in castigating the concept of MIPS and preserving the mystical nature of the performance of their machines (nobody can beat us if we keep our performance definitions secret, or sufficiently vague).

The aggregate measurement problem is not unique with aggregate MIPS. The same problem exists, in a hidden form, with the use of ITR. Aggregate MIPS implies a set of instructions executed in certain proportions; so does a benchmark that generates an ITR measure.

Furthermore, with a high level language benchmark, one has to worry about the instruction mix at two levels: the higher language level and the compiled object code level. These are often different due primarily to differences in compiler efficiencies. One also has to worry about any differences in the operating environment, since many modern languages are not just a simple compiler and a run stream. In Ada, for example, one is benchmarking more than just a compiler; a total operating environment is benchmarked.

A recent report from Intermetrics Corporation concerning benchmark results on their Ada compiler versus IBM'S Ada compiler and DEC's Ada compiler states the following for execution time performance [11]:

- Intermetrics' MVS Ada COMPILERS VERSUS IBM's Ada COMPILER
  - All tests were run on an IBM 3084
  - Of the 42 test compiled 37 ran successfully for both compilers
  - On the 37 successful tests
    - IBM was faster on 2 tests (by 30%)
    - Intermetrics was faster on 35 tests
    - Average ratio was 2.47:1 [In favor of Intermetrics]
    - Intermetrics 3 times faster or more on 8 out of 37 tests

- Intermetrics' Ada (running on IBM 4341) VERSUS DEC's Ada (running on VAX 11/780)
  - Intermetrics was faster on 39 of 52 tests (75%)
  - Average ratio was 3.7:1 (Favoring Intermetrics)

The results from this report serve to demonstrate the difference in performance on the same machine that can exist given the same set of high language instructions (in this case Ada) and a different set of machine language instructions (due to the difference in compiler efficiencies).

The report states that the Intermetrics to DEC ratio of 3.7:1 is "Much more than the variation in MIPS" [12]. The report does not specify what model IBM 4341 was used; however, assuming that the 4341 used was a model 12, the aggregate MIPS ratio would be 1.4 (1.5/1.06). It is possible, but not likely, that the difference between the performance ratio (3.7) and the MIPS ratio (1.4) is due to the Intermetrics compiler being that much more efficient than the DEC compiler.

2.3.2 The Definition Problem

One of the major problems with the use of MIPS could be quite easily solved by vendor cooperation. As has already been pointed out MIPS implies an instruction mix. Hence, for a precise comparison using MIPS one needs to know exactly what instruction mix went into the making of a quoted MIPS estimate.

Vendors often complain about the imprecision of MIPS; yet they will not take simple and effective steps toward increasing the precision of MIPS estimates. If vendors would publish MIPS estimates for their processors along with the exact instruction mix used in the estimate, a planner could use MIPS with as much precision as he could estimate the instruction content of his projected workload, and this could be estimated with considerable accuracy in many cases. Why will most vendors not do this? Partly because of a HIDDEN-AGENDA problem which will be discussed in section 2.3.4.

It is easier to understand the use and significance of MIPS if it is interpreted as a measure of CPU speed and not as a measure of CPU power. Power is the ability to do work. More precisely, work performed in unit time equals power. To illustrate why MIPS should be considered a measure of CPU speed and not of CPU power consider the following equation (EQUATION 2) published in an article discussing MIPS [17].

EQUATION 2:

$$\text{MIPS} = \frac{1}{\text{(Cycles per average instruction)*(Cycle time in microseconds)}}$$

An examination of the above formula reveals that it is composed of two variables. MIPS is functionally (inversely) dependent upon 1), the cycle time of the processor and upon 2), the number of cycles it takes to execute the average instruction.

The cycle time of the CPU is usually fixed for a given processor. Cycles per average instruction depend to a large extent upon the architecture of the computer. The important thing to keep in mind is that a computer instruction is the entity that performs work for a user. Individual Computer instructions vary in the amount of work which they are capable of producing. And they may exhibit large variance across computer architectures. Hence one machine may be turning 15 Millions Instruction per Second (MIPS) and doing the same 'work' for the user that another machine would require 30 Million instruction per second. MIPS then are best thought of as speed of the CPU, and not power. We are likely to see more of this sort of variance in instruction efficiency in the near future as the concept of the Reduced Instruction Set Computer (RISC) matures. RISC computer systems are based on the concept of optimizing a small set of instructions that are frequently used, and avoiding instructions that are not frequently used. This approach has the goal of reducing hardware complexity and increasing hardware speed.

A RISC processor as opposed to the conventional Complex Instruction Set Computer (CISC) processor has three commonly accepted characteristics:

1) a RISC machine must execute one instruction in a single clock cycle

2) a RISC machine must use a fixed format for instructions

3) a RISC machine must use only a load/store architecture for interacting with memory

An examination of EQUATION 2 will reveal that for a processor with a given clock speed (cycle time in microseconds), one can increase the MIPS rate by reducing the average number of cycles per instruction. Given current technology, RISC computers can be expected to run at higher MIPS rates than comparable CISC computers. However, the power of the individual RISC instruction may not be as great in some cases as the power of the individual CISC instructions; hence, MIPS comparisons between RISC and CISC machines may be subject to large discrepancies. It should be noted that significant architectural differences make the use of unadjusted MIPS suspect.

2.3.3 The Power Problem

There is a real problem and a hidden-agenda problem with the use of Millions of Instructions Per Second (MIPS) as a measure of power for a central processing unit (CPU). However, MIPS numbers are useful for planning and estimating purposes if a few simple caveats are assumed.

The real problem with MIPS as a measure of relative CPU power is the fact that it does not actually measure power (i.e. work done in unit time), even though people still use it that way. It seems reasonable, therefore that one would like to understand some concept of computer "Power" measurement and where MIPS would fit into this concept.

In order to legitimately measure power, one must first define work. Units of software work can be defined as that which takes place when a byte of data is transferred from a processor to a storage device. In other words, the central processor does a unit of work on main storage for every byte transferred into real storage. This type of definition of Software Work is the basis for what is often called Software Physics.

> The definition of software work proposed by Kolence in the context of his software physics [see Kolence (1972)] is very similar from a pragmatic viewpoint to the one given by Rozwadowshi: 'a processor performs one unit of software work on some storage media when one byte of that media is altered.' [18]

In Software Physics, a workload or workload component is characterized by the work done by the CPU, DASD, terminals, printers, and other devices as they process and move data about the system. The resulting workload description is usually called a Software Work Vector.

To understand where MIPS fits into the concept, consider the operation of a sawmill in the following analogy:

The purpose of a sawmill from the viewpoint of the owner is to process as many logs through the mill during hours of operation as he possibly can. The power of the sawmill can be measured by the quantity of lumber that the mill will output in some given unit of time (e.g.- one day). The maximum rate at which a sawmill could turn out lumber (throughput) would correspond to the notion of computer power in the computer.

The speed at which the saw spins (Revolutions Per Minute or RPMs) would correspond roughly to MIPS in a computer system. Obviously, the speed of the saw is important. In fact, if one assumes the speed of the feeder belts, the diligence of those placing and removing logs from the belts, the diameter and sharpness of the saw are all equal, then the speed of the saw becomes the determining factor in the "power" of this uni-sawmil.

The REAL PROBLEM, then, with the use of MIPS is that SOME people may forget that they are talking about "the speed of the saw", and that the other factors in sawmills being compared may not be close enough to allow using "the speed of the saw" as a good comparison tool.

This is exactly the case when trying to use MIPS to compare the performance of a Microprocessor to that of a well equipped mini or mainframe. Although the micro may have a very fast saw, the feeder mechanisms, in comparison to a mainframe, simply are not enough to give it comparable power in terms of the ability to achieve work in a throughput sense.

> How does the 80386 compare to other processors, small and large? Clearly, the 80386 is a microprocessor because of its implementation on a single chip. But do its characteristics qualify it as a mainframe? Certainly its speed is more than adequate for that classification. But while the architecture is as complex as some mainframes, it is incomplete in the area of I/O paths....
>
> Given a proper I/O design and adequate memory size and speed, the 80386 would compare closely in performance to an IBM Model 4341-1 or 4341-2, depending on a number of factors. The higher instruction execution rate of the 80386 would be offset by the more powerful instructions of the 4341. Memory and I/O bandwidth would also have to be evaluated for specific designs. In capability, however, the 80386 could perform at the level of an IBM 4341. The major differences in performance would depend on how the 80386's I/O subsystem compared to the 4341's I/O subsystem and the use of special 4341 instructions. [19]

You do not use "the speed of the saw" to compare the power of two sawmills UNLESS it is reasonable to assume that the other essential mechanisms are similar in performance capability. The same statement is true with MIPS.

2 3.4 The Hidden-Agenda Problem

Let us now discuss a HIDDEN-AGENDA PROBLEM. To continue our sawmill analogy, the fact that some people may incorrectly use "the speed of the saw" to compare two sawmills, does not mean that everyone does, or that "the speed of the saw" is a useless measure. It is time that we stopped being amused whenever some vendor defines MIPS as standing for Meaningless Indicator of Processor Speed. The simple fact that some people do not understand the use or limitation of MIPS as a way to estimate comparative power between computers does not mean that all people do not understand.

Many vendors are in opposition to the use of MIPS for rating their computers (fearing that those who do not understand the limitations may misuse it). However that reason (fear) does not necessarily invalidate the use of MIPS by other people. Aspirin and penicillin can be misused if their limitations are not understood but certainly no one would argue their use be discontinued for that reason.

Similar objections can be raised to single evaluation scores in any environment. The fact is, we are often faced with either using a single, numerical estimate; or not quantifying at all. For example, in no justifiable sense can a single I Q. score measure the total potential of a human being. Yet, for those who understand the limitations, a tested I.Q. score may be useful for some academic predictions.

Should educators stop using I.Q. scores because some people may not understand their limitations? Should we stop using MIPS simply because the uninitiated may not understand that the usefulness of MIPS as an estimator of power becomes very tentative when you try to use it across non-homogenous computers or with non-homogenous workloads?

Most of those who use MIPS to estimate relative computer power are VERY, VERY aware that when you cross vendor lines the usefulness of MIPS decreases drastically. However, if "other" factors are reasonably close, MIPS can at least put you in the right ballpark, which is usually where you want to get when you use MIPS __ in the ballpark!

A vendor's objection to the use of MIPS as a rough measurement of relative CPU power is, it would seem, a HIDDEN-AGENDA PROBLEM. The vendor often does not want the user to independently estimate the power of his computer. Why? There cannot be much market differentiation based upon power or MIPS, unless your processor is definitely superior in power to its competitors. The vendor wants you to buy, based upon the features he has selected to market. For the most part, he does not market raw MIPS. He markets a differentiated, positioned product. MIPS is an indignity; it reduces a vendor's product to a single number. Ouch! A MIPS number puts everyone on a similar basis. Let's face it, if a MIPS number was the most accurate and dependable single number measurement in the world, most vendors would not like it; and they cannot be faulted for feeling this way. But that does not mean that the number should not be used; especially if one understands the usefulness and limitations.

SECTION 3

TEST RESULTS

3.1 MITRE NOMAD2 BENCHMARK RESULTS

In 1985, as preparation for support of upcoming NASA procurements, the author tested an initial version of a sizing benchmark on a series of IBM compatible mainframes. Among the mainframes tested were IBM's 3083EX, 3083JX, 3081KX, and 3090/200; Amdahl's 5840, 5850, 5860, 5867, 5870 AND 5880; and NAS's 9050 and 9060 processors.

The benchmark used was coded using NOMAD2 (data base management system and registered trademark of D&B Computing) and was implemented by running concurrent Virtual Machines under IBM's VM/CMS operating system. The rules for successful implementation of the benchmark favored an environment with minimum constrained I/O. This meant that the vendor was encouraged to supply enough DASD and memory so that I/O was never a bottleneck. Thus a simple way to eliminate I/O wait time was chosen so that the processors would always closely approach 100% busy.

The NOMAD2 benchmark performed four NOMAD2 routines sequentially: 1) a database dump, 2) a database load, 3) a series of change requests against the database, and 4) a list request (report) against the database. The database consisted of about 3000 records with each VM/CMS user possessing and exercising against his own copy of the database. The benchmark provided heavy loading. One VM/CMS virtual machine running the NOMAD benchmark on a dedicated IBM 3083EX processor required 80 seconds of CPU time, 100 seconds of DASD service time (3380 non-cache), and issued 7288 I/O requests.

Using the NOMAD2 benchmark and the chosen workload, the Internal Throughput power of the tested processors was measured and tables of ITR values for the processors tested were compiled.

TABLE 1 shows ratios of ITRs for the various processors compared to the IBM 3083EX. The data was based on single CPU service times for one user. The ratios would have to be multiplied by an appropriate constant to project the ITR for the dyadic processors involved. A projection multiplier of 1.8 was used as a conservative estimate based upon previous benchmarking experiences.

TABLE 1

ITR RATIOS

	NO.		PUBLISHED MIPS FROM
PROCESSOR	CPUs	PROJECTED ITR	CMI CORP.
AMDAHL 5840	1	1.78	7.4
AMDAHL 5850	1	2.41	9.8
AMDAHL 5860	1	3.12	12.40
AMDAHL 5867	2	4.25	15.40
AMDAHL 5870	2	5.47	21.00
AMDAHL 5880	2	5.36	21.70
NAS 9050	1	2.17	8.10
NAS 9060	1	2.73	10.20
IBM 3083 EX	1	1.00	4.06
IBM 3083 JX	1	2.36	8.12
IBM 3081 KX	2	4.16	15.40
3090/200	2	7.00	27.70

STATISTICS FOR THE ABOVE DATA:
Correlation between MIPS and ITR is (0.996)
Regression equation: [MIPS=(3.93)*(ITR)-0.254]
Standard error of the estimate is (0.62)

NOTE: A dyadic processor has two CPUs that share memory and one set of I/O channels. A dual processor has two separate sets of I/O processors The AMDAHL 5870 ran as a dyadic processor, while the AMDAHL 5880 ran as a dual processor. All other two CPU systems listed above are dyadic.

Linear regression analysis was performed on the data in TABLEs 1 through 5. In TABLE 1, published MIPS from CMI Corporation were correlated with projected ITR ratios from the tested computers. The correlation was high positive (0.996) and demonstrates that published MIPS are a useful predictor of projected ITR across the span of computers tested given the test workload.

TABLE 2 presents all possible ratios between MIPS and ITR for each processor listed in TABLE 3. The purpose of this table is to show that the MIPS ratio between two individual machines in the test group is a useful predictor of the projected ITR between the same two machines. For example, if one were planning to upgrade from an IBM 3083EX to an AMDAHL 5870, one would look at the MIPS (Y line) ratio and find 5.17. One would use 5.17 to estimate the CPU power ratio between the 5870 and the 3083EX. The projected ITR between the two machines is 5.47. In this case our estimate (5.17) is 5.5% under the projected ITR (5.47).

TABLE 2

PROJECTED
ITR AND MIPS
RATIOS

		5840	5850	5860	5867	5870	5880	9050	9060	83EX	83JX	81KX	3090 /200
ITR	X	1.78	2.41	3.12	4.25	5.47	5.36	2.17	2.73	1.00	2.36	4.16	7.00
MIPS	Y	7.40	9.80	12.40	15.40	21.00	21.70	8.10	10.20	4.06	8.12	15.40	27.70
5840	X		1.36	1.76	2.39	3.08	3.02	1.22	1.54	0.56	1.33	2.34	3.94
	Y		1.32	1.68	2.08	2.84	2.93	1.09	1.38	0.55	1.10	2.08	3.74
5850	X	0.74		1.29	1.76	2.27	2.23	0.90	1.13	0.41	0.98	1.73	2.91
	Y	0.76		1.27	1.57	2.14	2.21	0.83	1.04	0.41	0.83	1.57	2.83
5860	X	0.57	0.77		1.36	1.75	1.72	0.70	0.87	0.32	0.76	1.33	2.24
	Y	0 60	0.79		1.24	1.69	1.75	0.65	0.82	0.33	0.65	1.24	2 23
5867	X	0.42	0.57	0.73		1.29	1.26	0.51	0.64	0.24	0.56	0.98	1.65
	Y	0.48	0.64	0.81		1.36	1.41	0.53	0.66	0.26	0.53	1.00	1.80
5870	X	0.32	0.44	0.57	0.78		0.98	0.40	0.50	0.18	0.43	0.76	1.28
	Y	0.35	0.47	0.59	0.73		1.03	0.39	0.49	0.19	0 39	0.73	1.32
5880	X	0.33	0.45	0.58	0.79	1.02		0.40	0.51	0.19	0.44	0.78	1.31
	Y	0.34	0.45	0.57	0.71	0.97		0.37	0.47	0.19	0.37	0.71	1.28
9050	X	0.82	1.11	1.44	1.96	2.52	2.47		1.26	0.46	1.09	1.92	3.23
	Y	0 91	1.21	1 53	1.90	2.59	2.68		1.26	0.50	1.00	1.90	3.42
9060	X	0.65	0.88	1.14	1.56	2.01	1.97	0.80		0.37	0.87	1.52	2.57
	Y	0.73	0.96	1.22	1.51	2.06	2.13	0.79		0.40	0.80	1.51	2.72
83EX	X	1.78	2.41	3.12	4.25	5.47	5.36	2.17	2.73		2.36	4.16	7.00
	Y	1.82	2 41	3.05	3.79	5.17	5.34	2.00	2 51		2.00	3.79	6.82
83JX	X	0.75	1.02	1.32	1.80	2.32	2.27	0.92	1.16	0.42		1.76	2.97
	Y	0.91	1.21	1.53	1.90	2.59	2.67	1.00	1.26	0.50		1.90	3.41
81KX	X	0.43	0.58	0.75	1.02	1.32	1.29	0.52	0.66	0.24	0.57		1.68
	Y	0 48	0.64	0.81	1.00	1.36	1.41	0.53	0.66	0.26	0.53		1.80
3090	X	0.25	0.34	0.45	0.61	0.78	0.77	0.31	0.39	0.14	0.34	0.59	
/200	Y	0.27	0 35	0.45	0.56	0.76	0.78	0.29	0.37	0.15	0.29	0.56	

STATISTICS FOR THE ABOVE DATA:
Correlation between MIPS RATIOs and ITR RATIOs is (0.9938)
Regression equation: [MIPS RATIO=(0.9672)*(ITR RATIO)+0.0346]
Standard error of the estimate is (0.1215)
AVERAGE (Y) DIFFERENCE IN MIPS AND ITR RATIOs is 6.7%
MAXIMUM (Y) DIFFERENCE IN MIPS AND ITR RATIOs is 21.1%

The correlation between all ratio pairs of 0.9938, a standard error of the estimate of 0.1215, and an average percent absolute difference of 6.7% between the MIPS ratios and the ITR ratios verifies that published MIPS figures are a useful predictor of projected ITR for the tested processors running the test workload (the benchmark).

TABLE 3 shows ITR ratios for the tested processors based upon the results of running the NOMAD2 benchmark on 20 concurrent VM/CMS virtual machines. In all cases the 20 concurrent users pushed the target machines to at least an average 95% CPU busy during the total benchmark run. Hence this table is an estimate of ITR that takes into consideration the concerns expressed by Andrew Lockey of AMDAHL [9]. Published MIPS from CMI Corporation were correlated with measured ITR from the tested machines where the tested machines were actually operated very close to saturation. The correlation was again high positive (0.963) and demonstrates that published MIPS are a useful predictor of measured ITR across the test group running the test workload.

TABLE 3

MEASURED ITR

PROCESSOR	NO. CPUs	ITR RATIOs	PUBLISHED MIPS FROM CMI CORP.
AMDAHL 5840	1	1.76	7.40
AMDAHL 5850	1	2.34	9.80
AMDAHL 5860	1	2.87	12 40
AMDAHL 5867	2	4.55	15.40
AMDAHL 5870	2	5.09	21.00
AMDAHL 5880	2	4.81	21.70
NAS 9050	1	2.11	8.10
NAS 9060	1	2.67	10.20
IBM 3083 EX	1	1.00	4.06
IBM 3083 JX	1	2.38	8.12
IBM 3081 KX	2	4.47	15.40
3090/200	2	5.81	27.70

STATISTICS FOR THE ABOVE DATA:
Correlation between MIPS and ITR is (0.963)
Regression equation: [MIPS=(4.39)*(ITR)-1.13]
Standard error of the estimate is (1.81)

NOTE: A dyadic processor has two CPUs that share memory and one set of I/O channels. A dual processor has two separate sets of I/O processors. The AMDAHL 5870 ran as a dyadic processor, while the AMDAHL 5880 ran as a dual processor. All other two CPU systems listed above are dyadic.

The purpose of TABLE 4 is to show that the MIPS ratio between two individual machines in the test group is a useful predictor of the measured ITR between the same two machines. For example, if one were planning to upgrade from an IBM 3083EX to an AMDAHL 5870, one would look at the MIPS (Y line) ratio and find 5.17. One would use 5.17 to estimate the measured ITR ratio between the two machines. The measured ITR ratio between the 5870 and the 3083EX is 5.09. In this case our estimate is 2% under the measured ITR ratio.

TABLE 4

PROCESSOR COMPARISONS USING ITR AND MIPS RATIOS

													3090
		5840	5850	5860	5867	5870	5880	9050	9060	83EX	83JX	81KX	/200
ITR	X	1.76	2.34	2.87	4.55	5.09	4.81	2.11	2.67	1.00	2.38	4.47	5.81
MIPS	Y	7.40	9.80	12.40	15.40	21.00	21.70	8.10	10 20	4.06	8.12	15.40	27.70
5840	X		1.32	1.63	2.58	2.89	2.73	1.19	1 52	0.57	1.35	2.53	3.29
	Y		1.32	1.68	2 08	2.84	2.93	1.09	1.38	0.55	1.10	2 08	3.74
5850	X	0.76		1.23	1.94	2.18	2.06	0.90	1 14	0.43	1 02	1.91	2.49
	Y	0.76		1.27	1.57	2 14	2 21	0.83	1 04	0.41	0 83	1 57	2.83
5860	X	0.61	0.81		1.58	1.77	1.68	0.73	0.93	0.35	0.83	1.56	2.02
	Y	0.60	0.79		1 24	1.69	1 75	0 65	0 82	0 33	0.65	1 24	2 23
5867	X	0.39	0.51	0.63		1.12	1.06	0.46	0.59	0.22	0.52	0.98	1.28
	Y	0.48	0 64	0 81		1.36	1 41	0.53	0.66	0 26	0.53	1 00	1.80
5870	X	0.35	0.46	0.56	0.89		0.94	0.41	0 53	0.20	0.47	0.88	1 14
	Y	0.35	0.47	0.59	0.73		1.03	0 39	0 49	0.19	0 39	0 73	1 32
5880	X	0.37	0.49	0.60	0.94	1.06		0.44	0 56	0.21	0.49	0.93	1.21
	Y	0.34	0 45	0.57	0.71	0.97		0 37	0.47	0.19	0.37	0.71	1.28
9050	X	0.84	1.11	1.36	2.16	2.42	2.28		1.27	0.47	1.13	2.12	2.76
	Y	0 91	1 21	1 53	1 90	2 59	2 68		1 26	0.50	1.00	1.90	3.42
9060	X	0.66	0.87	1.07	1.70	1.90	1.80	0.79		0.37	0.89	1.67	2.17
	Y	0 73	0.96	1.22	1.51	2.06	2 13	0.79		0 40	0.80	1 51	2 72
83EX	X	1.76	2.34	2.87	4.55	5.09	4.81	2 11	2 67		2.38	4.47	5.81
	Y	1.82	2.41	3 05	3.79	5 17	5.34	2 00	2 51		2 00	3.79	6.82
83JX	X	0.74	0.98	1.21	1.91	2.14	2.02	0.89	1.12	0.42		1.88	2.44
	Y	0 91	1.21	1.53	1.90	2.59	2.67	1 00	1 26	0.50		1 90	3.41
81KX	X	0.39	0.52	0.64	1.02	1.14	1.08	0.47	0.60	0.22	0.53		1.30
	Y	0 48	0.64	0.81	1 00	1.36	1.41	0 53	0 66	0.26	0 53		1.80
3090	X	0.30	0.40	0.49	0.78	0.88	0.83	0.36	0.46	0.17	0.41	0.77	
/200	Y	0.27	0 35	0.45	0.56	0.76	0.78	0 29	0 37	0.15	0 29	0 56	

STATISTICS FOR THE ABOVE DATA:
Correlation between MIPS RATIOs and ITR RATIOs is (0.974)
Regression equation: [MIPS RATIO=(1.0396)*(ITR RATIO)-0.0229]
Standard error of the estimate is (0.2505)

AVERAGE (Y) DIFFERENCE IN MIPS AND ITR RATIOs is 13.1%
MAXIMUM (Y) DIFFERENCE IN MIPS AND ITR RATIOs is 40.7%

The correlation of 0.9735 with a standard error of the estimate of 0.2505 and the average percent absolute difference of 13% between the MIPS ratios and the tested ITR ratios verifies that published MIPS figures are a useful predictor of tested ITR for the target processors given the test workload (benchmark).

TABLE 5 compares the results of TABLEs 1 and 3. TABLE 1 presents projected ITR, based upon the CPU required for one VM/CMS machine running on a dedicated processor. TABLE 3 presents an actual measure of ITR ratios (3083 Ex = 1) when the tested processors where pushed to the limit. The purpose of this table is to show that there is a high positive correlation between projected ITRs and measured ITRs (0.978). However, in most cases, the projected ITR is larger than the measured ITR.

TABLE 5

MEASURED ITR RATIOS
VS
PROJECTED ITR RATIOS

PROCESSOR	NO. CPUs	PROJECTED ITR RATIOs	MEASURED ITR RATIOs	RATIO OF MEASURED TO PROJECTED
AMDAHL 5840	1	1.78	1.76	0.9887
AMDAHL 5850	1	2.41	2.34	0.9709
AMDAHL 5860	1	3.12	2.87	0.9199
AMDAHL 5867	2	4.25	4.55	1.0706
AMDAHL 5870	2	5.47	5.09	0.9305
AMDAHL 5880	2	5.36	4.81	0.8974
NAS 9050	1	2.17	2.11	0.9724
NAS 9060	1	2.73	2.67	0.9780
IBM 3083 EX	1	1.00	1.00	1.0000
IBM 3083 JX	1	2.36	2.38	1.0085
IBM 3081 KX	2	4.16	4.47	1.0745
3090/200	2	7.00	5.81	0.8300

STATISTICS FOR THE ABOVE DATA:
Correlation between:
PROJECTED ITR RATIO and MEASURED ITR RATIO is (0.978)
Regression equation: [TESTED ITR=(0.843)*(MAX ITR)+0.396]
Standard error of the estimate is (0.3024)

NOTE: A dyadic processor has two CPUs that share memory and one set of I/O channels. A dual processor has two separate sets of I/O processors. The AMDAHL 5870 ran as a dyadic processor, while the AMDAHL 5880 ran as a dual processor. All other two CPU systems listed above are dyadic.

A good conservative rule of thumb for using MIPS to predict ITR for sizing purposes (where you wish to upgrade from a smaller to a larger machine) is to take the MIPS ratio between the target and current machines and reduce it by 20 percent. The 20 percent reduction will cover most of the cases where your estimate is too small, and will add a comfortable buffer in others cases.

3.2 ANALYSIS OF REPORTED BENCHMARK DATA

3.2.1 Single-Function MIPS Numbers

Let's examine some single-function MIPS numbers. Dr. David S. Lindsay of National Advanced Systems (NAS) has developed a set of benchmarks to measure CPU speed. Dr. Lindsay has run his selected set of 117 tests on various CPUs [13],[14]. Table 6 contains results from some of Dr. Lindsay's tests (IBM plug-compatible machines plus DEC). The results show the variability of various machine level operations across several different vendor's products and contains comparative performance figures in microseconds.

TABLE 6

INSTRUCTION TIMES (MICRO-SECONDS)
AND PUBLISHED MIPS

COLUMN	1	2	3	4	5	6	7	8	9	10
CPU	I-4 ASSN	LARGE ARRAY	3-D ARRAY	INNER LOOP	SUBR C-R	I-4 ADDS	I-4 MULT	R-8 ADDS	R-8 MULT	CMI MIPS
AMDAHL 5860	0.07	0.45	0.82	0.33	1.80	0.05	0.17	0.17	1.34	12.40
AMDAHL 5890	0.05	0.40	0.40	0.22	0.93	0.02	0.10	0.05	0.10	17.22
IBM 3081-KX	0.13	0.90	1.18	0.49	2.19	0.06	0.33	0.17	0.39	8.12
IBM 3090/200	0.09	0.72	0.59	0 24	1.20	0.03	0.14	0.06	0 11	15.40
IBM 4341-12	0.52	2.72	8.48	2.18	10.07	0.43	2.30	0.87	4.23	1.50
IBM 4361-5	0.52	4.10	8.94	2.05	9.39	0.35	3.20	2.60	2.05	1.33
IBM 4381-2	0.32	2.65	4.00	1.29	6.24	0.25	0.99	0.51	0.37	2.70
NAS XL80	0.04	0.22	0.39	0.29	1.12	0.02	0.05	0.03	0 04	25.00
VAX 11/780	1.23	7.52	7.66	2.89	15.80	0.94	2.53	2.93	4.98	1.06
VAX 8600	0.33	1.64	2.12	0.99	3.68	0.21	0.71	0.45	1.11	4.40
VAX 11/785	1.45	2.67	5.13	2.03	13.25	0.62	1.54	1.81	3.22	1.70

Dr. Lindsay developed his benchmarks solely to measure CPU speed. And in those cases where the processors were multi-processors, the speed of a single CPU was measured. Hence MIPS estimates shown are for the base (single) processor on those system that are dual-processor system.

TABLE 7 is based upon the data of TABLE 6, the times have been inverted and expressed as rates in single-function MIPS. I have labeled the columns from 1 to 10 for ease of reference. The columns contain performance times for 1) assignment or movement of data in memory (I4 ASGN), 2) loading a large array that will not fit in most cache memories, hence a "cache buster activity" (LARGE ARRAY); 3) loading a 3-dimensional array (3-D ARR); 4) branching and looping from DO Loops (INNER LOOP); 5) making subroutine calls and returns (SUBR C-R); 6) performing an integer Load and Add (I4 ADDS); 7) performing in integer load and multiply (I-4 MULT); 8) performing a real number Load and adds (R8 ADDS); 9) performing a real number load and multiply (R8 MULT); 10) MIPS number taken from the latest chart published by CMI Corporation. In the case of dual processors, the MIPS rating of the base processor was used.

TABLE 7

INSTRUCTION RATES (MIPS)
AND PUBLISHED MIPS

COLUMN	1	2	3	4	5	6	7	8	9	10
	I-4	LARGE	3-D	INNER	SUBR	I-4	I-4	R8	R-8	CMI
CPU	ASSN	ARRAY	ARRAY	LOOP	C-R	ADDS	MULT	ADDS	MULT	MIPS
AMDAHL 5860	14.29	2.22	1.22	3.03	0.56	20.00	5.88	5.88	0.75	12 40
AMDAHL 5890	20.00	2.50	2.50	4.55	1.08	50.00	10.00	20.00	10.00	17.22
IBM 3080-KX	7 69	1.11	0.85	2.04	0.46	16.67	3.03	5.88	2.56	8.12
IBM 3090/200	11.11	1.39	1.69	4.17	0.83	33.33	7.14	16.67	9.09	15.40
IBM 4341-12	1.92	0.37	0.12	0.46	0.10	2.33	0.43	1.15	0.24	1.50
IBM 4361-5	1.92	0.24	0.11	0.49	0.11	2.86	0.31	0.38	0.49	1.33
IBM 4381-2	3.13	0.38	0.25	0.78	0.16	4.00	1.01	1.96	2.70	2.70
NAS XL80	25.00	4.55	2.56	3.45	0.89	50.00	20.00	33.33	25.00	25.00
VAX 11/780	0.81	0.13	0.13	0.35	0.06	1.06	0.40	0.34	0.20	1.06
VAX 8600	3.03	0.61	0.47	1.01	0.27	4.76	1.41	2.22	0.90	4.40
VAX 11/785	0.69	0.37	0.19	0.49	0.08	1.61	0.65	0.55	0.31	1.70

Dr. Lindsay emphasizes the anomalies that are exhibited by the test data.

> ... We have pointed out several performance anomalies of the machines we have tested, anomalies that could easily dominate CPU consumption for some benchmarks or applications. We hope that publishing these data will help analysts avoid such pitfalls in the future.
>
> Because of these performance anomalies, we have avoided averaging the test results and deriving some kind of "overall MIPS" or "power" figure. As was pointed out in the introduction, and as has become clear from the data presented above, relative performance varies over a wide range depending on the instruction mix. Even among machines with similar architecture, large differences occurred. [15]

MIPS as generally used implies an instruction mix. Benchmarks also involve the use of instruction mixes, often at two levels, a higher language level and the compiled (machine code) level. If one wishes to avoid the use of instruction mix instruments, one must not only avoid the use of MIPS, but also the use of benchmarks which attempt to map a real workload, for real workloads are comprised of instruction mixes.

That there is variance in the test data is understandable. Variance in the performance of machines with different speeds is absolutely to be expected. The sort of variance that Dr. Lindsay points out is the variance in the ratio of the execution times for different functions across different machines. Dr. Lindsay uses as one of his illustrations the performance of the NAS XL-80 versus the VAX 11/785. Dr. Lindsay states that:

> ...thus the assignment test measures cache speed. The cache speeds of the machines tested varied considerably. The fastest was the NAS XL-80, and the slowest of those tested was the DEC VAX 11/785; they differed by a factor of 36 (a much larger spread than the supposed "MIPS" ratings of the two machines would suggest). [16]

An examination of TABLE 7 will reveal what Dr. Lindsay is talking about. The ASSN rate for the NAS XL-80 is 25 and the ASSN rate for the VAX 11/785 is 0.69. Hence the ratios between ASSNs would be 36 (25/0.69). The MIPS ratio between the XL-80 (one of its processors) and the VAX 11/785 would be 24 (25/1.06).

But this sort of isolated comparison does not tell the whole story. Take a few minutes and examine the results for the VAX computers in TABLE 7. One can see that the 1.7 MIPS rating for the VAX 11/785 is suspect in view of the test data. No single measurement for the VAX 11/785 expressed as a MIPS number is equal to or greater than 1.7. The closest number to the 1.7 is the value 1.61 in column 6 which represents integer load and add performance. In the case of the other two VAX entries, only one entry in each line is equal to or greater than the MIPS estimate. In each case, it is the value in column 6 (integer load and Add).

On the other hand, four measurements for the XL-80 (assignment, integer load and add, real time load and add, and real time multiply) are equal to or greater than the MIPS number. If one is going to measure MIPS, then speed of assignment (not just load and add) should probably weigh heavily in the estimates. Apparently Dr. Lindsay would agree:

> One important set of tests in this benchmark measures assignment time, the time required to move data from one location to another--clearly a measure of great importance. In fact, MacDougall (Ref. 5) has found that fully 50% of the instructions executed by production COBOL jobs on IBM systems merely move data. This result is probably not unique to either COBOL or IBM systems. If the single most important CPU performance measure were to be selected, it would probably be assignment time. The assignment statements that we timed were the four-byte integers (called INTEGER*4 in FORTRAN).
>
> Because assignment time is so important, we have used it as a basis to compare other CPU functions. [16]

It appears that in the case of the VAX processors, the load add values would have had to be used almost exclusively to derive a MIPS number equal to the estimates used. It is not surprising, then, that the ratio between the MIPS estimate for the XL-50 and the VAX 11/785 is less than the ratio between their corresponding ASSNs.

When you ask which of two machines is faster, the answer may depend upon whether or not you wish to compare the speed of specific machine functions or whether you wish to compare an aggregate speed of some sort. Look again at TABLE 7 and examine the rates for the VAX 11/785 and the IBM 4361-5.

You will see that the 4361-5 is 2 8 (1.92/0.69) times faster on ASSNs than is the 11/785; however, the 11/785 is 2.1 (0.65/0.31) times faster on I-4 MULT than is the 4361-5. Asking which machine is faster is much like asking which of two track teams is faster. Depends upon which race they are running. We do not however, hesitate to pick a winner at a track meet using a composite result of speeds (finishes) from their individual races.

To argue that we cannot pick the faster of two computers, or the faster among several computers, would be analogous to arguing that we cannot pick a winner at a track meet simply because different teams are faster in different events. We can and must make choices between entities based upon composite differences. We do it every day. If Dr. Lindsay were to manage a track meet perhaps he would announce at the conclusion of the day's events that "We have avoided adding the race results and hence have not selected an "Overall winner" for the track meet." I realize full well that computing is not track. There are cases where we need to select a computer for a very specialized performance; however, there are many more cases where we do, indeed, need to selected a computer for a general (hence composite) performance. Knowing all the individual instruction rates in the world will be of no help in the overall evaluation of competing processors without the method and the resolve to construct an appropriate aggregate evaluation device.

Sometimes, of course, one computer is simply faster in every category than another. Look in TABLE 7 and compare the entries between the AMDAHL 5890/200 and the IBM 3090/200. You will see that the AMDAHL 5890/200 is faster in every category than the IBM 3090/200. This, however, represents an easy case and in such situations performance comparisons are relatively easy.

3.2.2 Regression Analysis and Weighted Instruction Rates

Multiple regression analysis was performed on the data presented in TABLE 7. The first multiple regression that was performed included the eight IBM plug-compatible processors listed in TABLE 7, in addition to four selected instruction rates along with the published MIPS numbers. The instruction rates that were selected were:

1) assignment of data in memory (column 1: I-4 ASSN)

2) branching and looping from DO loops (column 4:INNER LOOP)

3) integer load and add (column 6: I-4 ADDS)

4) real number load and adds (column 8: R-8 ADDS)

TABLE 8 shows the results of the multiple regression analysis performed upon these variables (8 processors, 4 instruction rates, and the published MIPS number).

The coefficient of multiple correlation of 0.9986 (TABLE 8) establishes quite well that when the four variables considered are taken as a set, they form a useful predictor of the corresponding MIPS number. However, the question that needs to be addressed is how well the variables would predict MIPS if they were given a 'realistic' workload weight.

TABLE 8

RESULTS OF MULTIPLE REGRESSION CORRELATION
FOR 8 PROCESSORS

Actual VS Predicted

Observation Number	Actual	Predicted	Difference	%Difference
1	12.40000	12.49815	-0 09815	-0.79156
2	17.22000	17 51085	-0.29085	-1 68904
3	8 12000	7.13145	0 98855	12.17427
4	15.40000	15.45532	-0 05532	-0.35923
5	1.50000	1.74559	-0.24559	-16.37288
6	1.33000	1.19428	0.13572	10.20449
7	2.70000	3.21678	-0.51678	-19.14005
8	25.00000	24.91757	0.08243	0 32973

Regression Equation:

$Y=-0\ 4881229+0.5640717*X_1+2.867562*X_2+-0.3588757*X_3+0.5806958*X_4$

Coefficient of Determination is (0.9972268)

Coefficient of Multiple Correlation is (0 9986)

STD Deviation of Estimate is (0 6897138)

To generate TABLE 9, postulated instructions weights were assigned to each of the four instruction categories used in TABLE 8. TABLE 9 shows the aggregated instructions rates calculated from the separate instruction rates and corresponding weights; in addition, all possible ratios between the aggregated rates for each of the processors and the corresponding published MIPS numbers are shown. The weights selected for each of the categories were:

1) 50% for assignment (I-4 ASSN)

2) 15% for branching and looping (INNER LOOPS)

3) 20% for integer load and add (I-4 ADDS)

4) 15% for real number load and adds (R-8 ADDS).

TABLE 9 shows that across the eight IBM plug-compatible computers, published MIPS numbers are a useful predictor of the performance of the machines executing a 'realistic' workload instruction mix. The Correlation for all the pairs of ratios is 0.9855 (TABLE 9). One of the other pertinent statistics is the average percent difference in the ratios across all the pairs. This average was based upon the average absolute difference. The average difference of 13% (TABLE 9) with a maximum difference of 38% (TABLE 9) shows that the MIPS ratio between two machines (across the eight compared) is a useful predictor of the performance ratio of the same two machines (given the postulated instruction mix).

Obviously other weights (workloads) could be justified; however, Dr. Lindsay is probably correct in his opinion of the importance of the assignment function [16].

TABLE 9

WEIGHTED INSTRUCTION RATES VS PUBLISHED MIPS WITHOUT DEC PROCESSORS

COMPUTER		5860	5890	81-KX	3090	4341	4361	4381	XL-80
RATE 1:		14.29	20.00	7.69	11.11	1.92	1.92	3.13	25.00
WEIGHT 1:		0.50	0.50	0 50	0.50	0.50	0.50	0.50	0.50
RATE 2:		20.00	50.00	16.67	33.33	2.33	2.86	4.00	50.00
WEIGHT 2:		0.20	0 20	0.20	0.20	0.20	0.20	0.20	0.20
RATE 3:		5.88	20.00	5.88	16.67	1.15	0.38	1.96	33.33
WEIGHT 3:		0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
RATE 4:		3.03	4.55	2.04	4.17	0.46	0.49	0.78	3.45
WEIGHT 4:		0.15	0 15	0.15	0.15	0 15	0.15	0.15	0.15
AGGREGATE									
RATE:		12.48	23.68	8.37	15.35	1.67	1.66	2.78	28.02
MIPS	Y	12.40	17.22	8.12	15.40	1.50	1.33	2.70	25.00
5860	X		1.90	0.67	1.23	0.13	0.13	0.22	2.24
	Y		1.39	0.65	1.24	0.12	0.11	0.22	2.02
5890	X	0.53		0.35	0.65	0.07	0.07	0.12	1.18
	Y	0.72		0.47	0.89	0.09	0 08	0.16	1 45
3081	X	1.49	2.83		1.83	0.20	0.20	0.33	3.35
	Y	1.53	2.12		1.90	0.18	0 16	0.33	3 08
3090	X	0.81	1.54	0.55		0.11	0.11	0.18	1.83
	Y	0.81	1.12	0.53		0.10	0.09	0.18	1.62
4341/12	X	7.49	14.20	5.02	9.20		1.00	1.66	16.80
	Y	8.27	11.48	5.41	10.27		0.89	1.80	16.67
4341/5	X	7.51	14.25	5.03	9.23	1.00		1.67	16.85
	Y	9.32	12.95	6.11	11.58	1.13		2.03	18 80
4381/2	X	4.50	8.53	3.01	5.53	0.60	0.60		10.09
	Y	4.59	6.38	3.01	5.70	0.56	0.49		9.26
XL/80	X	0.45	0.85	0.30	0.55	0.06	0.06	0.10	
	Y	0.50	0.69	0.32	0.62	0.06	0.05	0.11	

```
Correlation for all XY ratios (r) is (0.9855)
Regression equation for XY RATIOS: Y=(0.9982*X)+(0.0300)
Standard error of the estimate is (0.7513)

AVG X =3.0534                  SD (X)=4.3720
AVG Y =3.0780                  SD (Y)=4.4284

AVG ABS DIFFERENCE=0.3850      AVG (Y) DIFFERENCE=13.22%
SD  ABS DIFFERENCE=0.6457      SD  (Y) DIFFERENCE=10.00%
MAX ABS DIFFERENCE=2.7224      MAX (Y) DIFFERENCE=38.00%
```

The second multiple regression that was performed on the data from TABLE 7 included all eleven processors listed, and the four selected separate instruction rates along with the

MIPS estimate. TABLE 10 shows the results of the multiple regression analysis performed upon these variables (eleven processors, four instruction rates, and the published MIPS number).

TABLE 10

RESULTS OF MULTIPLE REGRESSION CORRELATION
FOR 11 PROCESSORS

	Actual	VS	Predicted	
Observation Number	Actual	Predicted	Difference	%Difference
1	12.40000	12.48550	-0.08550	-0.68953
2	17.22000	17.42226	-0.20226	-1.17457
3	8.12000	7.26150	0.85850	10.57264
4	15.40000	15.56545	-0.16545	-1.07438
5	1.50000	1.99833	-0.49833	-33.22220
6	1.33000	1.43508	-0.10508	-7.90088
7	2.70000	3.44757	-0.74757	-27.68761
8	25.00000	24.90757	0.09243	0.36971
9	1.06000	1.05906	0.00094	0.08903
10	4.40000	3.92895	0.47105	10.70579
11	1.70000	1.31873	0.38127	22.42785

Regression Equation:

$Y=-0.2006487+0.5457429*X_1+2.860474*X_2+-0.3632949*X_3+0.592882*X_4$

Coefficient of Determination is (0.9969736)

Coefficient of Multiple Correlation is (0.9985)

STD Deviation of Estimate is (0.5783098)

The coefficient of multiple correlation of 0.9985 (TABLE 10) establishes quite well, in this case also, that when the four variables considered are taken as a set, they form a useful predictor of the corresponding MIPS number across IBM and DEC processors. Again, the question that needs to be addressed is how well the variables would predict MIPS if they were given a 'realistic' workload weight.

TABLE 11 shows that across the eleven listed computers (eight IBM plug-compatible machines plus three DEC machines) published MIPS numbers are _not_ a very useful predictor of the performance of the machines executing a 'realistic' workload instruction mix. The Correlation for all the pairs of ratios is down to 0.9182 (TABLE 11). The other pertinent statistic is the average percent difference in the ratios across all the pairs.

TABLE 11

WEIGHTED INSTRUCTION RATES VS PUBLISHED MIPS WITH DEC PROCESSORS

										11/		11/
COMPUTER		5860	5890	81KX	3090	4341	4361	4381	XL-80	780	8600	785
WEIGHT 1:		14.29	20.00	7.69	11.11	1.92	1.92	3.13	25.00	0.81	3.03	0.69
RATE 1:		0.50	0 50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
WEIGHT 2:		20.00	50.00	16.67	33.33	2.33	2.86	4.00	50.00	1.06	4.76	1.61
RATE 2:		0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
WEIGHT 3:		5.88	20.00	5.88	16.67	1.15	0.38	1.96	33.33	0.34	2.22	0.55
RATE 3:		0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
WEIGHT 4:		3.03	4.55	2.04	4.17	0.46	0.49	0.78	3.45	0.35	1.01	0.49
RATE 4:		0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
AGR RATE:		12.48	23.68	8.37	15.35	1.67	1.66	2.78	28.02	0.72	2.95	0.82
AGR MIPS:		12.40	17.22	8.12	15.40	1.50	1.33	2.70	25.00	1.06	4.40	1.70
5860	X		1.90	0.67	1.23	0.13	0.13	0.22	2.24	0.06	0.24	0.07
	Y		1.39	0.65	1.24	0.12	0.11	0.22	2.02	0.09	0.35	0.14
5890	X	0.53		0.35	0.65	0.07	0.07	0.12	1.18	0.03	0.12	0.03
	Y	0 72		0.47	0.89	0.09	0.08	0.16	1.45	0.06	0.26	0.10
81KX	X	1.49	2.83		1.83	0.20	0.20	0.33	3.35	0.09	0.35	0.10
	Y	1.53	2.12		1.90	0.18	0.16	0.33	3.08	0 13	0.54	0 21
3090	X	0.81	1.54	0.55		0.11	0.11	0.18	1.83	0.05	0.19	0.05
	Y	0.81	1.12	0.53		0.10	0 09	0.18	1.62	0.07	0.29	0.11
4341	X	7.49	14.20	5.02	9.20		1.00	1.67	16.80	0.43	1.77	0.49
	Y	8.27	11.48	5.41	10.27		0.89	1.80	16.67	0.71	2.93	1.13
4361	X	7.51	14.25	5.03	9.23	1.00		1.67	16.85	0.43	1.78	0.50
	Y	9.32	12.95	6.11	11.58	1.13		2.03	18.80	0.80	3.31	1.28
4381	X	4.49	8.53	3.01	5.52	0.60	0.60		10.09	0.26	1.06	0.30
	Y	4.59	6.38	3.01	5.70	0.56	0.49		9.26	0.39	1.63	0 63
XL-80	X	0.45	0.85	0.30	0.55	0.06	0.06	0.10		0.03	0.11	0.03
	Y	0.50	0.69	0.32	0.62	0.06	0.05	0.11		0.04	0.18	0.07
11/780	X	17.32	32.87	11.61	21.30	2.31	2.31	3.86	38.89		4.10	1.14
	Y	11.70	16.25	7.66	14.53	1.42	1.25	2.55	23.58		4.15	1.60
8600	X	4.23	8.02	2.83	5.20	0.56	0.56	0.94	9.49	0.24		0.28
	Y	2.82	3.91	1.85	3.50	0.34	0.30	0.61	5.68	0.24		0.39
11/785	X	15.17	28.78	10.17	18.65	2.03	2.02	3.38	34.04	0.88	3.59	
	Y	7.29	10.13	4.78	9.06	0.88	0.78	1.59	14.71	0.62	2.59	

```
Correlation for all XY ratios (r) is (0.9182)
Regression equation for X Y RATIOS: Y=(0.5771*X)+(0.6860)
Standard error of the estimate is (1.8473)
AVG X  = 4.2390                    SD (X) = 7.4220
AVG Y  = 3.1324                    SD (Y) = 4.6647
AVG ABS DIFFERENCE= 1.4515         AVG (Y) DIFFERENCE= 36.65%
SD  ABS DIFFERENCE= 3.5187         SD (Y)  DIFFERENCE= 35.29%
MAX ABS DIFFERENCE=19.3366         MAX (Y) DIFFERENCE=184.08%
```

The average difference of 37% with a maximum value of 184% shows that the MIPS ratio would not be a useful predictor of the performance ratio of two machines (given the postulated instruction mix) when you span IBM, AMDAHL, NAS, and DEC. If one examines the aggregate rate row one will see why the inclusion of the DEC results brought the correlation down (from 0.9855 to 0.9182). If the aggregate rates are taken as reasonable estimates of MIPS, one can see that the VAX processors are seriously overstated.

SECTION 4

CONCLUSIONS

The results of the analyzed test data clearly show (with correlations above 0.97) that for a given instruction set such as IBM compatible, there is an excellent correlation between published MIPS and benchmark data in a complex data base environment such as NOMAD2 running under VM/CMS. This is a counter example to the generally held belief that MIPS are meaningless.

If other words, if one wishes to estimate the internal throughput performance of an IBM compatible mainframe where the I/O environment is not constrained, then published MIPS numbers are a very useful predictor of internal throughput performance.

The results of the analysis also show that published MIPS correlate well (0.986) with aggregated instruction rates. The average percent difference in the corresponding ratios of MIPS and the aggregated instructions rate (13.22%) is approximately equal to the average percent difference in the ratios of MIPS to measured ITR (13.1%). The analysis also showed, however, that when DEC processors were considered along with the IBM plug compatibles, the average percent difference in MIPS and aggregate instruction ratios became quite large (36.7%).

These results simply serve to verify the accuracy and sensibility of using MIPS as a capacity planning number when your environment is homogenous (in this case IBM compatible) and when you are aware that you will have to engineer suitable I/O and memory subsystems to support whatever processing power you put into place.

One must not place undue confidence in a MIPS number across vendor lines. Also, one must not place undue confidence in a MIPS number if one is trying to predict delivered power in support of non-homogenous workloads. This would include the case of comparing two identically configured processors from the same vendor running different operating systems (such as IBM'S VM/CMS and MVS). One faces the probability that two different operating systems will result in the generation of different machine level instruction mixes even if the higher level load is identical.

Comparing published MIPS figures for microprocessors, minis, and mainframes may serve no useful purpose because a micro, a mini, and a mainframe do not have comparable I/O and memory support mechanisms. If one can not feed the sawmill, or get the cut lumber off the receiver tray, then it does no good to make the blade turn faster. This paper has presented evidence that published MIPS for IBM plug-compatibles and published MIPS for DEC are inadequate for predicting the actual performance differences of the two classes of machines.

The use of MIPS as an estimator becomes dangerous when one forgets what he is trying to estimate. With MIPS, the sensible planner is trying to get a ballpark estimate of internal throughput power of a processor operating in a non I/O constrained environment across reasonably similar processors with postulated similar workloads. If a planner remembers these caveats, then a MIPS estimate can be and usually is very useful.

REFERENCES

[1] Dzelzgalvis, I., "Performance Comparison Measures for Computer Systems," Computer Performance Evaluation Users Group; ed. by James E. Weatherbee,National Bureau of Standards Special Publication 500-52, pp. 163-164 (1979).

[2] Wadsworth, E., "A Systematic Approach to Obtaining Better Online Response Time," CMG '86, International Conference on Management and Performance Evaluation of Computer Systems, pp. 831 (December 1986).

[3] Smith, R.D., "Vector Processing and the Performance Analyst," CMG '86, International Conference on Management and Performance Evaluation of Computer Systems, pp. 740 (1986).

[4] Hockney, R. W., Jesshope, C. R., Parallel Computers, Adam Hilger Ltd, Bristol, Great Britain (1981).

[5] Hockney, R. W., Jesshope, C. R., Parallel Computers, Adam Hilger Ltd, Bristol, Great Britain, pp. 51 (1981)

[6] Hockney, R. W., Jesshope, C. R., Parallel Computers, Adam Hilger Ltd, Bristol, Great Britain, pp. 56 (1981)

[7] Hockney, R. W., Jesshope, C. R., Parallel Computers, Adam Hilger Ltd, Bristol, Great Britain, pp. 123 (1981).

[8] Jordan, H.F, "HEP Architecture, Programming and Performance", Parallel MIMD Computation: HEP Supercomputer and its Applications, ed. by J.S. Kowalik; MIT Press, Cambridge, Massachusetts, pp. 1-40 (1985).

[9] Lockey, A., "Large Systems Performance Studies in VM/CMS Environments," CMG '86, International Conference on Management and Performance Evaluation of Computer Systems, pp. 217-219 (1986).

[10] MacKay, A. F., Arrow's Theorem, The Paradox of Social Choice, Yale University Press, New Haven, Connecticut (1980).

REFERENCES (Concluded)

[11] IR-MA903, Generated Code Performance: Benchmark Test Results for Intermetrics' IBM Real-Time Ada Compilers, Intermetrics Inc. (May 1987).

[12] IR-MA-903, Generated Code Performance: Benchmark Test Results for Intermetrics' IBM Real-Time Ada Compilers, PIWG RESULTS (INTERMETRICS VS. DEC), Intermetrics Inc. (May 1987).

[13] Lindsay, D. S., "CPU Performance Comparisons Among Vendors," CMG '86, International Conference on Management and Performance Evaluation of Computer Systems, pp. 179-186 (1986).

[14] Lindsay, D. S., Bell, T. E., "Directed Benchmarks for CPU Architecture Evaluation," CMG '86, International Conference on Management and Performance Evaluation of Computer Systems, pp. 379-384 (1986).

[15] Lindsay, D. S., "CPU Performance Comparisons Among Vendors," CMG '86, International Conference on Management and Performance Evaluation of Computer Systems, pp. 183,184 (1986).

[16] Lindsay, D. S., "CPU Performance Comparisons Among Vendors," CMG '86, International Conference on Management and Performance Evaluation of Computer Systems, pp. 179 (1986).

[17] Serlin, O., "MIPS, DHRYSTONES, and Other Tales", DATAMATION, pp. 112-118 (June 1986).

[18] Ferrari, D., Computer Systems Performance Evaluation, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, pp. 237 (1978).

[19] Well, P., "Inside the IBM PCs, Intel's 80386 Architecture,", BYTE, Volume 11, Number 11, pp. 98 (1986).

DISTRIBUTION LIST

MITRE

A10

E. L. Key
A. J. Tachmindji
C. A. Zraket

D11

B. M. Horowitz

D12

J. J. Fearnsides
S. W. Gouse

D22

Technical Report Center (2)

D110

E. S. Herndon
D. A. MacQueen

D115

W. N. Bays
S. A. Bell
K. B. Hennigan
W. P. Kincy
T. R. Mitchell
F. M. Richards
J. K. Richardson (25)
Systems Engineering (15)

D116

M. A. Allen (Gren)
D. M. Erb
E. E. Hill (NAHQ)
S. I. Linde
L. W. Morgan
P. A. Nussman (NAHQ)
J. V. Pietras (NAHQ)
J. F. Spitzer
R. W. Zears

D117

J. S. Brown
P. M. Brown
P. J. Gregor
R. M. Jackson
W. B. Wood

W120

A. H. Ghovanlou
L. L. Holmes
H. M. Strong

W121

A. A. Elsaway
K. H. Miller
T. R. Mitchell
A. D. Zeitlin

W122

F. L. Willingham

DISTRIBUTION LIST (Concluded)

W107

Records Resources (2)

NASA

F. Abernethy
J. Arnold
D. Bell
W. Bennett
S. Berthiaume
B. L. Brady
J. Broadfoot
J. Cools
R. Dorman
J. Garman
D. Ingram
J. Jue
S. Milo Keathley
L. Kirbie
C. Krpec
S. Leathers
E. McHenry
C. Mains
D. Simanton
J. Smith
W. Stewart
B. Stuckey
P. Stull
R. Voigt
C. Vowell
G. Walker
D. L. Ward - FD4
Technical Library (3)