APt
PRESS

Box 378, Pleasantville \Y 105?0 (914)769-3323

BUI,K RA TE

U.S. Postage Paid

PERMIT NO. 28 Pleasantville
NY Io570

APL NEWS

NOIICES

As of rJune 1., the new buslness address of APL Press will b as shown on the cover page. Sales tax will no longer appl to orders from Pennsylvania, but will apply to orders from
New York.

Professor James England presented a paper on the use of APL in courses in Calculus and in Linear Algebra at the ,January meeting of the American Mathematical Society.

Copies of the L2-page paper are available on request from the author at Swarthmore College, Swarthmore, PA 19081.

The followinc is a brief extract from the introduction:
ln the experimental courses we use computing in a central

way: to represent the ideas and concepts of calculus and linear algebra and facilitate their analysis and explor- ation. To do this we use APL as the notation. ln doing so we use the advantages of the APL notation to alter the presentation of a number of topics contained in calculus

and linear algebra. The advantages we see in the APL notation are its relative conciseness, its lack of ambi- guity, its ability to handle arrays ln a stralght forward

manner and its demand for explicitly defined functions.
A result of uslng the APL notation in these courses is a hlgh degree of integration of the subjects of llnear
a lgebra and calculus.
APL Press publications will be displayed at two forth-
com i ng confe rences :

0ntar lo Universlties Computing Conference University of Waterloo, Waterloo, 0ntario, l4ay l0-31-
Conference on Computers in the Undergraduate Curricula lvlichlgan State University, East Lansing, Ml, June 19-22

o Professor R. L. Spence is currently exploring the u.e of U

APL in teaching a f4asterrs level course in computer-aided electronic circuit design, and invites correspondence from interested parties. His address ls: lmperlal College of
Science and Technology, Department of Electrical Engineer- ing, Exhibition Road, London SW7 2BT, Ensland.
The recent announcenent of Dr. Spencets election as a Fellow of the IEEE cites him asrr,., a keen proponent of
the APL notationrr and as author of a textbook which uses
APL as the excluslve mathematical rrotation (Resistive Circuit Theorv).
o Jeff Shallit has been too busy with the work of hls sopho-

more year at Princeton to produce a PLAY column for this lssue. However, he submlts a correction to the CANON function given in the last issue ([3] z+(^\,tlAtt t)/A) and
Copyr ight 1977' APT, Presg

a

vv

i
I

I

wishes to draw attention to the omission noted in the following letter from Robert A. Smith of Scientific Tine

Sha r ing Co rpo ra t ion :
A bit of history was omitted from the APL-PLAY column in

your Vol. 2, No. 1 issue. As some of your readers may recall, the Self-Reproducing Expression problem was pub-

lished in the Problems Section of the Vol. lr, No. 2, Jan-uary 1971 issue of APL Quote-Quad. The three solutions mentioned in your last issue can be found in the January and April 1971 Problems Sectlon column.
o The many readers who have indicated an interest in APL

implementations on micro-computers may expect to flnd sev- eral relevant articles in the July issue of Byte.

o Material for publication in the newsletter should be sub- mitted in clean draft form for consideration. Book manu- scripts should include a preface or introduction indicating

the nature and scope of the material.

BOOK 9aat4 9(.Qailea REVIEW 9yzo"ot" Qlniaezrily

Gerrit A. Blaauw, qigilgl Svstem lmplementation. Prentice-Hall, Englewood Cliffs, N.J., 1975.

This book is a study of the implementation of digital com- puters, by one of the chief architects of lBMrs 160 series. It gives the cleanest and most complete presentation of dlgital systems to be found ln the literature. As such it

begins with the architecture, proceeds to the implementation strategy and is completed in the realization. These three views of the structure at hand are taken in turn in the elaboration of the nine chapters and four appendices which compr i se the book.
Throughout APL is used as an essential tool in the present-ation of concepts and algorithms. APL is introduced piece-meal as needed and is'nicely summarized in Appendix A, For

those already familiar with APL this material may be easily skipped. For those less acquainted with APL, the breadth of

the topics covered and the fact that.the descriptlons of the hardware may be executed should provide ample motivation for learning APL. The APL used ls at the level of APL/160, but

there appears to be no reason why the material will not run directly or with mlnor modlfications on most APL systems.
The followinc observations were made whi le using this book as a text in a graduate level course, and they constltute more of an overview of the book than a critique of its
contents.

Chapter 1 covers a number of prel-iminary items includlng the introduction of background information about System t60 which forms the basis of the running example througho!t the -_-

text. Chapter 2 covers addition, and of course subtraction,! -
of twors complement numbers, Thi architecture of addltlon 'v

is introduced and from it a bit adder is derived. The ripple adder leads to the carry-predict adder and lts varlants. Unfortunately the function pRED1 of page 45 is not prlnted and the function PRED from page 4, is repeated. The correct version of PREDq as supplled to this revieyirer by Dr, Blaauw is g iven be low:
I1l

127 t3l t4l Isl t6l

lll

v C+G PR4DGS 'I
A PREDICTED CARRY FOR GRAUPS aF 5
C+(pT\p0
Cli4J+CINGS

eI ; 3 l+dI i 4]v?[; +]^dltcs Cli2f+Gli3)v (IL i37^GL i47r,t ^/Il t 3

cl i77+Gl i27v (?l i2)^ct is])v (^/IL i 2

2 3 47,CING5

dti0l-Gt i7f\t (Tl i17 ^Cl i2l) v (

^

/?t ; 1
1 2 3l,C[]41)v^/Tt; 7 2 3 47,CINCs

47 , CINGs
3l,Gt;41)v^/?[;
2f,Gli37)v(^/?lI
V

ln addition to the speed up obtained by replacement in the calculation of the carry path, the carry save adder is also of importance as it wi ll occur in later chapters. Chapter 3 deals with multiplication and after introduclng multiplication of positive integers, problems of dealing with signed integers lead to recoding the multiplier and speeding up the process, A variety of techniques are introduced offering a wide range of speeds. lvjMPy which performs two bit shifts appears in Iater examples. At this point Blaauw suggests that Chapter 4 can be skipped to take up Chapters 5 and 6, at least in part, and then one
may return to Chapter 4 which deals wlth division, once [, again a variety of methods are discussed at the implement- v

ation levels. for example, ONEDIV develops the quotient one bit at a time using non-restoring division on signed operands. Although there are more sophisticated techniques discussed in this chapter, 1NEDIv is the division method selected for elaboration as part of Chapters 5 and 6 in the development of the data pa th.

Chapters 5 and 6 are closely related in that the data path is covered in Chapter 5 and control of the execution box ls con-sidered in Chapter 6. The first two sections of Chapter 5, which develop a data path for add/subiract and multiply, may be covered and then one may move onto the first section from Chapter 6 whlch develops an implementation for horlzontal nicroprogram control of the data path. Attention can then be returned to the development of a common parallel data path and then to consider various alternatives in reduclng the number and/or width of the data paths. When attention ls

agaln returned to Chapter 6, the divider is used as an example of how control can be moved from the mlcroprogram directly to hardware. lmplementation strategles for redu-ing the width of the microprogram word leads to introduction of encodings in the fields of microprogram words and to vertical microprogram- ming. By the concluslon of these two chapters control of the execution box and its data path are considered down to the gate contro I leve I .

Chapter 7 turns to system control and the development of the l-box functions for instruction sequencing. Architecture,

implementation, and gate control programs are developed, and whi le exanples are restricted to register (RR) and register-

indexed storage (RX) formats, sufflcient detai I is given to permit understanding the system implenentation problems.

Chapter 7 concludes with coverage of topics of encoders,

decoders, and priority, including the least recently used (LRU) a lgor i thm.

Chapter 8 discusses storage including topics of modular con- struction of storage, one level memory, and problems of address translation, hierarchical nemory in the form of a cache, and look-aside rinemory for address translation. Topics of storage access, accessing data which is not aligned on a

memory boundary, and pipelining conclude the Chapter.

Chapter 9 deals with communlcation between peripheral units and the rest of the system over a channel and device adapters

which interface the peripherals to the channel. There is a

wide range of peripheral device types and speeds and often the same peripheral device may be interconnected to differ-

ent computing systems with widely differing characteristics.
We should not be surprised that much of this chapter remains at the architectural level.
Appendix B gives a two page summary of System 560 instructions
, tnd a one page APL description of the lntel 8080. Appendices

\"i and D give a summary of machine arithmetic and switching algebra respectively. The APL enthusiast wi.l I note a missing left (extra right) parenthesis on p.t3l and a mislabeled table

on p.352 (NAND is expressed incorrectly in APL).

The 142 problems distributed throughout the chapters are in- structive and expand upon the range of examples, For instance, ln the chapter on addiqion, adders for the PDP 11/45 and CDC 6600 are considered. The chapters on multiplication and division echo these cases as well as provide exercises on alternatives to the text strategies.

.

Chapter 5 has the Tl SN7418L as an example (problem 5-3, p.18L). Unfortunately there appears to be flaws ln the function 1l\4P74181 in that the data book from Tl indicates that functlon selection is on pins 3.\.,5.6 and, not as indi-

cated in line 3. Llne 18 is ln error as the output is on POST\TS 71 10 9l and not POSrlr2 11 10 91. Pin 12 ls a ground as indicated in the comment on line 26.

ln a simjlar manner the problems from ehapters 6 through 9

extend the topics covered by asking the reader to deveiop aids- in understanding or verifying earlier material, or io .-.. consider alternative lmplementations. There are five exteni sive revlew exercises, including the investigation of f f oatill/ point architecture, at the end of Chapter 9.. An errata sheet supplied to this revlewer by the author con- tains mostly typographical errors. These may mar the perfec- tion of the book in detailbut they should not detract from its practical use, Except for a missing close bracket at the end of line L of oPERATION on p.118 ApL functions appear to be

I i sted correct ly.

As the physical size of computing systens continues to shrink, and as we become more involved with the pitfalls of programmlng,

r.re need to be reminded how wonderfully complex a conputer is. Professor Blaauwrs book not only exposes those inner workings but also shows how complexity and cost may be traded for speed in the implementation.
ON DERIVATIVES
In Chapter 8 of his Calculus in a new kev (ApL press,

1976), 0rth presents a set of ApL functions (represented in about a dozen lines of type) which precisely foimalize the chain rule and the other rules for differeniiation commonly taught in calculus courses. When entered in an ApL computer these functions provide automatic differentiation of complex expresslons. They also provide a clear and complete exposi- tion of the differentiation process. The following excerpt i I lustrates the approach used:

8.2 A DERIVATIVE FUNCTION

We will now define a monadic function 4 such that: \-.,
The arguments of 2 are to be vectors of characters repre-
sent ing func t ionsi

lr

{
t

The values of 2 wi ll also be character vectors functions, The function represented by a value be equivalent to the derivative of the function by the corresponding aigument of D,
The function 2 is defined recursively in terms of the de- rivative rules. For example, consider the monadic scalar function represented by the character vector ,(1oo)+2oo!.

The derivative of this function can be produced in terms of the dyadic plus rule. Evidently, the derivative is identi- cal to the derivative of the function 1oo plus the derivative of 2od. Consequently it must be true that the function re- presented by the character vector

Ai(1od)+2odr

representing of4isto
represented

is equivalent to the function represented by the character
vector def ined as follows:

U

r(r,(D'1o0'),')+i,4'2od'

lf we are to use the derivative rules.ln the defi- nition of the function a, we must first define a function

which determines the appropriate rule to be applied in each

case. The selection of a derivative rule is based on the factorization of the function whose derivative is being produced. That ls, in selecting a derivative rule to be

applied to a function F we first determine a primary scalar functlon f and either one monadic scalar function c or two monadic scalar functions F and c such that either:
H u or E u

fG u (F u)fc u

ln the first case we apply the monadic f derivative rule and in the second case the dyadic f derivative rule.
ln either form of the factorization of a described by
the above identities, t is equivalent to an expression such that whenever this expression is evaluated, the primary

function f is evaluated last. Thus to determine an appro- priate derivative rule for a character vector that is an argument of 2, we can determlne the symbol (and its location) of a prlmary scalar function that could be evaluated last whenever the function represented by this character string is eva l ua ted,
ln any expression without redundant outside paren-
theses, as in (A+B), the only candidates for the last

evaluated function are those which are not imbedded within parentheses, For example, when evaluatine the expression

v

((A +B) *C \ -Ax (B +C)
the func t ion :
before *, which
x is eva luated
eva I ua ted . The
Lt+t (t

is evaluated before ' and + is evaluated
i s in turn eva luated before -. F ina I ly,
before -, so that - is the ldst function
function

DEPTH | +\(u=L-P\ -a ,- ri u.Ee
can be used to determine those primary functions in an ex-pression which are not imbedded in paFentheses. DTPTH is a monadic function whose arguments are character vectors and whose values are vectors of integers lndicating the depth of imbedding within parentheses of each character in the corres-pond ing argument. For example:
DEPTH | ((A+B)*C) -Ax(Brc)'
7 2 2 2227 7 7 0 0 011111

The characters at depth 0 can be produced as followsl
s+t ((A+B \ t,c) -Ax (B + c r t
(O=DEP?H S') / S

primary scalar functions at depth o can be produced as
lows :

((o=DEPIH S)^S€r+-x?*')/g

Among the prinary functlons in an

wi ll be evaluated last ls the one thest to the left. The index of character vector representi ng the

duced as fo I lows:

tr+I+((0=DEprM) ^S€

I +- x:* r) 11
9

stJl

U

The fol

exPress i on, the one that at depth o which is fur- this function in the

express ion can be pro-

The primary scalar functlons with which we wi l.l be concerned for the derivative algorithm are +/-:,x,:,*,o, and o, The index of the last evaluated functioh from thls list in an expression is produced by the monadic function
egg+r+- x !*eo ' pFt ((o=DEp?H u, nuepg€).1t
For examp le:

PF S
I
and

B+I+PF r(1oa)t2od'
,(1od) x2o(t'tfl