Meeting with IBM People at Los Alamos Sept 19, 1956

(Dunwell)

1. no final decision now
2. avoid numerical values (eg. no of registers, etc)

Word size: 60 bit divisible by variety of nos. original thought
suggest 64 bit as binary no.
ability to divide binary-wise
can work with portions of words more easily
possibility of floating pt. half wd.
(diff. no. of bits to represent exp. in 32 & 64 case)
- goal that every bit in memory can be addressed.

Indexing: Traffic over mem. bus system - one of most difficult probs.
- must be able to send nos. anywhere
- ~~but~~ parts must be separate for multiplexing
Possibility of separate mem. for index quantities ?
- 0.2 μs core mem. suggested - can get many more registers than using Transistors. (hard to get above 16 with Trsts)

In Memory certain operations should be done without going over gen. bus system. Some can be done as part of reg. mem. cycle.
Eg, counting in memory is needed. Wd in mem. is read out added 1 and read back in memory in one mem. cycle.
May be Possible to check against limit.

Indexing: In all problems - how is programmer or assembly system going to cope with system -

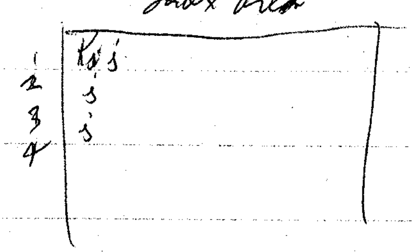
Example: of what human would write

R Add $\alpha$ (K)
subscript $j$ ($nj$)
subscript $j'$ ($j'$) ← $K + nj + j'$ type (need not avoid This Type)
Mpy by $A$
subscript $i - \frac{1}{2}$
subscript $j$

all nos. must be contained in indexable memory assumed.

$j$ as letter is interpreted to give address in which $nj$ is located,

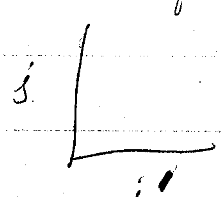
Index area



We want to be able to add in ~ 0.5 μs The indices $(K)+(nj)+(j')$

mpy is much harder

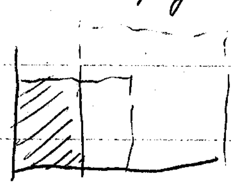
assume dimensions of ~~tables~~ tables are powers of 2 (remaining values can be set to 0.)



$K + nj + j'$

1101
101 0000
0011 000 0000
↑ last one need

Then we can mpy by off-setting

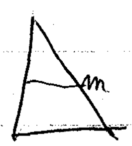


← can start new table overlapping

assume Index Reg contains

1. current value of subscript
2. # of bits of subscript ~~-~~ used as no. of shifts.
3. limit of itself; either no.

can increment index by any integer power of 2
and test against lim. m (which itself can be incremented)



2 schemes

1. for random access - use binary multiples of i's (space wasted)
2. for going thru arrays, - need direct adding other than 1.

---

Timing charts.
suggest example ~~o~~ Bagin's prob. inner 2 loops be worked out

If wd. more rapid arith? what about "precision values"? 4 categories e.g.
particularly mpy. & div. $\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$

or should opn have significance tag? not word itself.

(Lazarus)

Floating Point conclusions

1. Exponent:

(a) Sign + magnitude

(b) Mpy by $2^{\pm n}$ by logical add + subtr (fl pt shift)

2. Underflow

(two modes)

(a) special indication automatically goes to special routine { indicator operands counters preserved

(b) set to zero (with indicator ?)

3. Overflow.

(a) special indication (exclude $\frac{x}{0}$)

4. Zero:

(a) all zero fractions imply mathematical zero

$$\left(\begin{array}{l} \text{i.e. } x+0=x \\ 0\cdot x=0 \\ \frac{0}{x}=0 \end{array}\right) \qquad |x| \geq 0 \geq -|x|$$

Note zero detection facilities imply time saved.

5. Normalization:

(a) Ideal: take minimum time (no useless shifts) with no information loss. i.e. whenever possible shift left first

(b) Cost of implied unnormalized arithmetic.

(c) Division.

6. Double Precision:

(a) addition 6 times over single precision. actual 704

mpy 5 times

div - 6 times

(...chines) Tape & card input.
"magnetic cards"

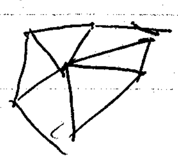
[tape units 100× faster than 704 tapes possible.
also more capacity. 100× or more 10⁷ wds.] External storage
cartridge loading

Data Inputs. mag. cards --
reading of typed material -- -- will be a year before we know how prog. is going.

Matrix Problems (Beckman)

1. Econometrics problem
System of simultaneous linear eqs.
Total output of steel industry = K. other industries.
675 order eqs have been done

2. Surveying problems
sum of angles should be 360°, etc



errors accumulate over large areas
Gauss theory of least squares.
Normal eqs:

$A A' =$ (matrix)(transpose of matrix) very specialized
largest 2400 or so.

3. Stress analysis.
large matrices mpy & inversion 200 or so are largest.

4. Regression analysis. linear.

- dep variable is assumed linear with indep. variables
  can take functions of indep. variable first if necessary
  large 50 variables 1000 observations.
  airforce Cambridge. 90 variables (weather prediction.)
  $(n \times m)(m \times n)$ type mpy.

5. Factor analysis

  statistics appl. to psychology (ref Thurstones "vectors of the Mind")
  eg. performance on a test = linear comb of mental abilities
  "Method of Principal components".

  to find roots & correlations. ~ 45, largest done so far
  of characteristic matrix,

6. Partial diff eqs.

- sometimes need to solve ~~[illegible]~~ systems.
- eg. implicit eqs. of elliptic type.
  finite no. of steps on single step solns.

7. Multigroup calcs.

  cannonical — Hotelling?
  etc

Most large matrices have many zero elements usually varies as cube of order.

Theory of von Neumann-Goldstine, Proc of Math Soc. - 1951

accuracy of matrix

depends on, $\left|\frac{\text{max eigenvalue}}{\text{min eigenvalue}}\right|$ "closeness ~~of~~ to singularity",

$A$ matrix

approx to inverse $A^*$

to solve $Ax = b$

$\underbrace{A^*A}_{\approx I} x = A^* b$

error: Term $I - AA^*$

use least upper bound $|I - AA^*|$

consider lengths of ~~a~~ vectors before & after transformation $Ax = y$

use ~~Statistical~~ arguments,

Results, matrix of order $n$ — $n$ = order of matrix.

if $n < .04\,\beta^{s/3}$ — $s$ = no. of characters in word (35)

$\beta$ = base of machine. (2)

then one can find an approx. to the inverse, better than the 0 matrix. $|I - A\cdot 0|$

$n < 130$ for 701,

actually 701 is good up to $n \approx 50$. to get a usable approx., (2 places or more)

takes about 1 hr on 701.

assumptions,

1, 701 is "well balanced" speed & word size wise

2. consider machine $r$ times faster

3. consider problem to be done in same time (e.g. ~~50~~ 1 hrs)

~~Q~~ What is size of matrix?

ratio of matrix size is $r^{\frac{1}{3}} : 1$ Then $\beta^{s/3} : 2^{35/3}$

gives $s = 35 + \log_2 r$

of $r = 1024$ $S = 45$ bits.

Floating point limit ?

- hard to say, but can use above for mantissa.

Storage requirements: goes up as sq rt

$$r^{\frac{1}{3}} : 1 = \sqrt{S} : \sqrt{2000}$$

$$S \simeq r^{2/3} (2000)$$

$$r = 2^9$$

$$S \simeq \underset{(64)}{2^6} (2000) = 128\,000.$$

---

use element + label (row, col. nos.)

zeros ~ 15 % of square matrix at least,

In finding characteristic roots,

Frames, etc. involve computing coefs. of characteristic eq. - can't do for large n's eg. $(x - \frac{1}{2})^n = 0$

max. coefs, ratio is $2^{143}$ for $n = 100$

use other methods. - make series of elementary transformations reduce to identity.

$$(R_m \cdots R_2 R_1) A = I$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & & \\ a_{n1} & \cdots & \end{bmatrix}$$

elimination procedure

try to use as big a division as possible.
rearrange rows & cols.

gauss Seidel method. for pos. def. sym. matrices, it converges.

$$\begin{cases} a_{11}x_1 + \cdots a_{1n} x_n = c_1 \\ \vdots \\ a_{n1} x_1 + \cdots a_{nn} x_n = c_n \end{cases}$$

$Ax = b$

can do this

$(A'A)x = (A'b)$

to make sym.

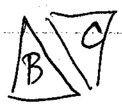
Start with approx for $x_n$'s.

Solve 1st eq for $x_1$

2nd eq for $x_2$

etc.

for $k^{th}$ approx.



$$x^k - x = \underbrace{\pm(B^{-1}C)}^{k}(x^0 - x)$$

if this → 0 rapidly it will converge rapidly

very slow for roots near 1.

very slow for $160 = n$ actual case.

$0 \leq |\lambda| < 1$

using tapes tape time may be $\frac{1}{3}$ of total or more.

no good method for roots & vectors of arbitrary matrices

Jacobi; von Neuman Goldstein, etc method for sym. matrices

Transform of $A$ $\quad TAT^{-1}$ $\quad$ preserves roots of $A$.

repeat $\ldots T_2 T_1 A T_1^{-1} T_2^{-1} \ldots$

Danielewski's method, Gibbons method ...

Jacobi's method: reduce matrix to diag form by above $T$'s,

$$\begin{pmatrix} \cos\theta & \sin\theta & & 0 \\ -\sin\theta & \cos\theta & & \\ & & \ddots & \\ 0 & & & \ddots \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & \\ a_{21} & & & \\ & & & a_{nn} \end{pmatrix} \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

choose $\theta$ to eleminate $a_{12}$ & $a_{21}$, also changes other elements in row 2 & col 2

needs to be done in high speed storage.
takes $n^3$ time.

gibbons method reduce to $\begin{pmatrix} \ddots & \ddots & & 0 \\ & \ddots & \ddots & \\ 0 & & \ddots & \ddots \end{pmatrix}$
3 stripes in one sweep
thru matrix. Then use other methods.

<u>good ref</u>:
$\boxed{\text{Forsythe May '53 Bull. Amer. Math. Soc}}$

---

Error Round-off

$$X_n - X = E(X_{n-1} - X)$$

we compute $\bar{X}_n = X_n + \epsilon_n$ due to round off

$$\bar{X}_n - X = \underbrace{E^n(X_0 - X)}_{\text{exact value}} + E^n\epsilon_0 + E^{n-1}\epsilon_1 + \cdots + \epsilon_n$$

upper bound of $\epsilon_i$'s

bound on $n$ is: $\dfrac{|I - E^n|}{|I - E|}|\epsilon|$

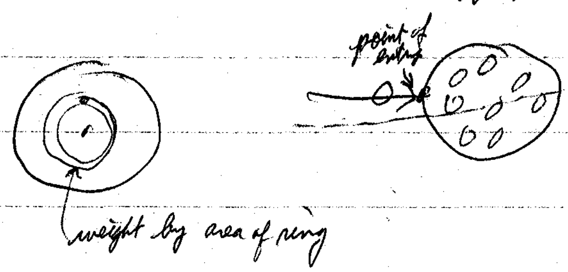
1. Indexing & addressing — Jane
2. word structure: significance, ½ wd arith, etc. — Max
3. auto progr. — Frank
4. vocabulary — Roger
5. Problems Time charts, — Gert
6. External memory, — Harwood
7. Input output — Ed.

2nd Day of Meeting

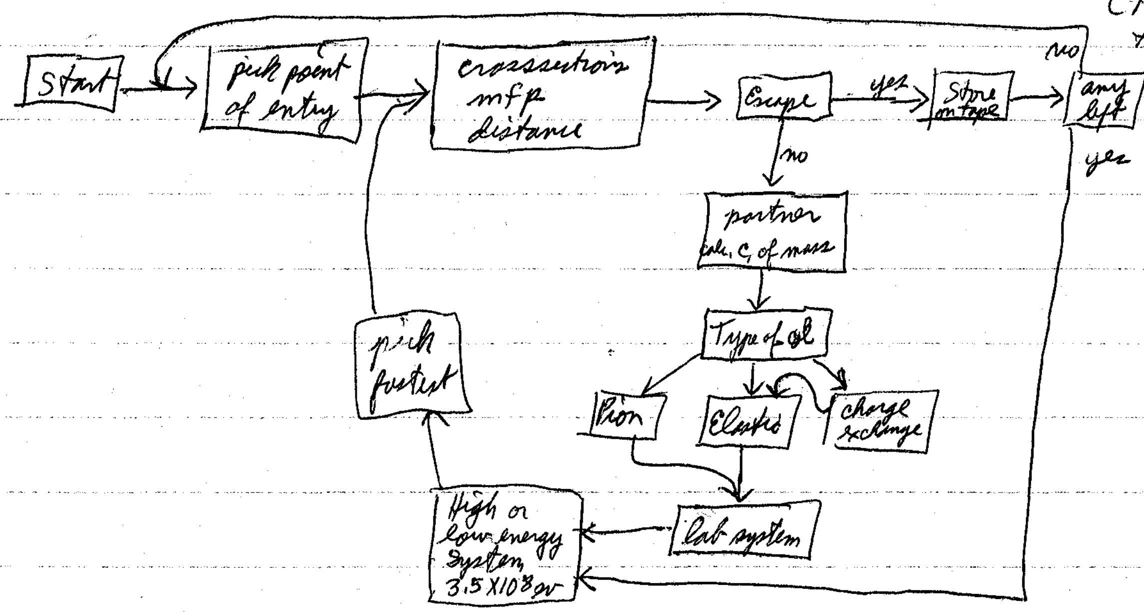
Monte Carlo Problem: (Bob Bivins)

– a "last resort" solution.

Example: high energy particle entering nucleus, if a collision occurs, energy is redistributed, – study what comes out, needs relativistic effects. 3 dimension, 2 pions can be given off in collision. Energy Range 6 Bev to 0,




{used multiple storing 6 quantities per word for crossect tables.




conversion back to lab system – 15 or more sqrts. cosines

random nos, "middle square" routine. have done

250 incoming particles / hr. at 3 Bev 1600 words per tape

1000 " " " at low energies. 40 tapes, store type, cos, energy

400 constants compressed into 96 wds.

used $A\cos^4\theta + B\cos\theta + 1$ stored A + B as 4 & 6 bits

12 msec for $\cos\theta$ calc. Stored {3 coordinates 7 bits

10 msec for $\sqrt{\ }$ about 1/3 3 component momentum 12-15 bits type

(can get around 15 particles coming out at 3 Bev for U) (recompute energy)

Time spent $\approx 1000$,
Time analysis

used 4 diff. analysis routines -
on same tapes,

half of $V^2$ were for calc sines from cosk.
debugging was very tough prefers floating point,

C of m calc took $\frac{1}{2}$ of time.
half words at higher speed - preferable.

Used 12 random nos per particle, not a long time factor
suggestion :- read in random no. tables from tape in blocks.
- separate serial mpyr.
- substitution by indirect addressing. from one table to ~~next~~.

$2^{B-2}$ degeneracy (720,000. cycle length on maniac I.)

1. collapse to zero.
2. cycling

Discussion of ½ wd - full wd :

assume 1. ½ floating mpy, div, add all 0.6 μs

2. (a) Instr wds ½ wds

(b) subscripts ¼ wds

(C) No instr or subscripts or data wd crosses mem wd. boundary

3. Data in ½ wds fl. pt.

4. reduced exp. length.

5. new instructions

suppress on zero (0.2 μs)

mpy or div by 2 (0.2 μs)

cumulative mpy. ?

6. ~~word size~~ is reflected in the instr.

7. half & full wd. instr can be mixed.

what is in opn. ?

| | | |
|---|---|---|
| example only | operation | 4 |
| | class | 3 |
| | FP | 1 |
| | absolute | 3 |
| | address | 15 |
| | Fast Mem | 1 |
| | Class of adr. | 2 |
| | Selector | 3 |
| | | 32 |

also need a tag to indicate indices.

a 0.1 Transfer $J_n^a$ to $J'$
b 0.2 ~~Transfer~~ Set $S_2^a$ to zero

a 1. Reset add $\alpha$ $S_{a,e}$
Sub $j$
Sub $j'$

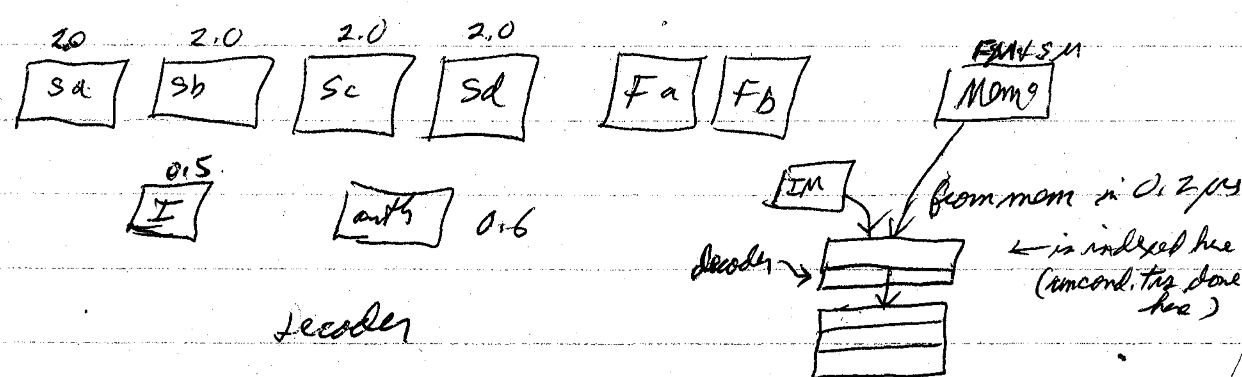
b 2. Mpy by A $S_{bd}$
Sub $i-\frac{1}{2}$
Sub $j'$

Inner loop **5.4**

a 3. add $S_2$ ~~Fa~~ $I_s$
b 4. Store $S_2$ ~~Fb~~ $I_s$
a 5. ($j-1$ and Test)
b 6. if not zero transfer to 1.

Note: one must mark fast loops to go in F.

7. Reset add $\Gamma$
Sub $i-\frac{1}{2}$
Sub $k$
8. Mpy by $S_2$
9. add $S_1$
10. Store $S_1$
11. ($k-1$ and Test)
12. if not zero transfer to 0.1
13. (Third loop)




decoder

3rd Day?

Large group will meet less than once a month.

- working groups will contact by phone once a week or more.

(B. Carlson)

| | |
|---|---|
| A | B being executed |
| B | D inst xtr |
| C | C (data xtr) |
| D | A (modified and returned) ← advanced for next time. |

suggest storing effective address

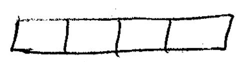
(Op)(label)(effective)

stored elsewhere.

(J. Woods.) Instr. Word structure

pieces of instr.

1. Instruction proper (operation)



about 4 catagories of break down (not complete micro prog)

Arith. opns { add, Mpy, Div — Div by, Div into } /alg { Fx, Fl

Logical

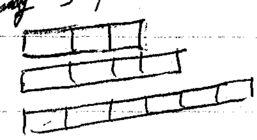
Transfers { conditional, etc — break points, etc?

comparisons

I/O opns (needs 4 levels)

=

many 3 field



(1) four fields about 4 bits ea. (16 ± 4 total)

- possible to have 2 words for some instrs, by leaving tag.
- Try to use same instr. format for I/O

~~###~~ address structure:

3 modes. (2 bits)

contains $X$ 1. immediate (actual operand in wd itself; ~~#~~ fixed constants less precision)

$L(X)$ 2. direct (address carried in order - for variable consts)

single indexed (in between)

$L^n(X)$ 3. indirect (blocks. multiple indexing - levels of addressing)

— 4. (no address) repeat type.

for 3. does one specify $n$ in instr, or does next step in chain control?

address Range.

(2) all of active memory requires 16 bits

" " ram " = 20 bits how done? I/O computer

Index Reg.

(3) ultrafast 4 (or more?) bits
fast mem 10 bits
slow mem 16

(8) but need 2 indx regs for many thing
consids. 9-1 = (8 or 10 bits more)

~~###~~

(4) accumulator Registers 4 bits (or more?)

to Here total 48 bits

I/O. ~~#~~ considerations
(5) a. bit address 6 bits
(6) b. field length 6 bits
(7) c. character length (bite size) 3 bits

total 63

(9) Breakpoints and/or selectors.

bit for opns which will go to subroutine where it is decoded. interpretive mode.

(Sweeney) arith operations

$(mod)\ A\ op\ ^{(m)}S \rightarrow {}^{(m)}A$ (or S) ?

Two types Add
Mpy
Div
Sub

modifier 4 classes
use sign
invert "
set +
set −

really need only one (m) with interchange order.

$(m)\ A\ op\ S \rightarrow A$

also do we need only one A or should we have 16 accumulators

Instrs like: − give exp. as another fl. pt. no.
− "give integer part",

orders should be reversible (symmetrical)
store address − Radd address

---

(Lazarus) Data Wd Format:

full wd. considered

8 bit $10^{\pm 38}$
9 bits $10^{\pm 77}$
10 bits $10^{\pm 154}$

The smaller the range the sooner one knows ~~[illegible]~~ trouble. (pick middle no.)

704 would be in balance at ~35 bits
so Stretch would need 7 more 42 bits fraction.

5 bits for selectors:

1 for zero ident.

4 for boundary tags.

half wd tags ?

precision indicator ?

large base ? $2^{12}$ can be used for 64

---

We should examine combination of opers. for background

for vocabulary

what arithmetic has been ~~[illegible]~~ found to be difficult?

what probs will be in future.

---

Dunwell: List of Topics

Performance Voorhees Griffith *

Instr Wd. format

Floating Point ~~[illegible]~~ * to be decided shortly.

Arith. Oper. *

Indexing & addressing. * review multiple indexing.

Data word format * inquire concerning 44 bits study verify

Auto. Prog. depends on above.

Input/output Use of - implications of large RAM & small mem,

also lot of EDPM considerations IBM

nwell) IBM Organization:

Labs: Poughkeepsie Prod Devel & Research
Endicott Prod Devel.
San ~~Jose~~ " "
Zurich
Watson at Columbia
Kingston Mil Products SAGE

Research at Pough

1. Physical Research group.
   - Solid state devices (magnetic, semicond, ferro-elec, super cond, etc)

2. Component Devel. group.
   - transfering to practical use (pilot plant)

3. Information Research group.

4. Advanced Planning group.
   Systems probs of future (10 years)

5. Stretch Organization
   (will move to Prod. Devel in about 12 mo.)
   a portion will remain in research

6. ~~Stretch~~

Within Stretch org.

Circuits 1. Devel. of circuits. (Transistors) (not fabrication directly) (PNP tran etc)
2. Core circuits.
3. Packaging group,
   - auto processes.

Design 4. Input Output Exchange (more groups on design will be activated later)
5.

Planning 6. Planning group (we will deal with)
7. Servicing group.

8. Group working with other customers