Feb 19, '58

Meeting at Los Alamos:

Carlson ✓
Wells ✓
Heising ✓
Carter ✓

Johnston ✓
Brooks ✓
Voorhees ✓
Lazarus ✓

Cocke ✓
Codd
Woods ✓
Frank ✓

Blaauw
Kolsky ✓

Test Prob: July 1 Target date to have all specified —
not in detail — may be premature
(Frank + Woods)

Blaauw: Command Set changes

Branch on bit Instr: — any loc in memory,

uncond branches 8 under Misc,

branch on bit is no-op if half wd.

performed in order: set to zero first then invert

- connect to accumulator only
- load with flags.
- byte size with connect — code conversion easier — inserts high order zeros.

P = progressive indexing

decimal, byte size = 4, connect opns byte size = 8

MR — register,

Now in core storage — Upcond + remainder 12, 13

J = 0 refers X16 I = 0 means no index

- "Store zero" new half wd store inst.
- Refill word in Memory (any wd) cond. or uncond. (a Misc. format)

Transmit + swap. full words

Transmit immediate: upto 32 half wds,

Swap immediate 16 full wds.

- Store address : direct format.

Rt. half of Transmit gives 18 bits only (not 19)

Store address doesn't work from accumulator — must put in index (any use X0)

- Rename (only now)   X0

- StoreV, C, R   half wd

Store Index   full wd (uses 18 bit addr.)

Floating Single length leaving B unchanged

- add to Memory Magnitute :

Integer with can say field is "unsigned" (pos.) in mem.
accumulator (Magnitude)

Fl Pt. is signed can ignore sign.   cannot make mem neg,

add same sign ?

---

no load double with flag ?

Store Instr. ctr & branch — any op code br.   takes care of (1.b pitfalls)

Rounding in Fl. Pt.

(Sweeney) — to get rid of bias not precision of extra bit.

would prefer rounded 47 bits than unrounded 48.

Incrementing functions, etc..

Carlson — can't think of case where non-round. in 704 hurt —

penalty give round
add 20%
mpy 10%
div <10%

Fl Div. 48 ÷ 48   does not assume norm.

[illegible] machine assumes normalized nos — penalty is on unnorm.

Fl Store into Acc   Sign → Sign bits reg.

Mpy not, add not

→ Divide single should be rounded if 10% penalty is all.

Divide double not.

Noisy Mode:

- add - preceding normalization bit 47 ← still want this. ←
- mpy? div? ← do before here also if most convenient.
  Most concerned with adds. others we use our judgment.
  put in where most convenient.
- zero results: (zero)-(zero) no noise.

Suppose: 0 in acc give an add order. { no fast add in this case / exp. of 0 counts }

question of -48 shift?

LA would prefer to omit shift.

i.e. assume 1 does lie just beyond end, - physical not math. zero.

Interrupt.

A. Interrupt on Branch Instruction
Which is the result of the branch

(1) What is in instr. counter

(a) Succesful branch (3 known) (① is in IC)

fix branches → (b) Instr. Read ① ② ③

(c) Address Invalid ① ③

(d) Index flag (Count + Br) ①

(1 step ~~[illegible]~~ Back (I-C))

(Things one might want to know:)

① where was branch?

② Where was branch to?

③ Was branch succesful?

(no one wd. fixup is possible)

(")

B. On full wd branch (Store IC)
Interrupt on Data write invsl address

~~[illegible]~~ -hrs. counting is done -

Q: if a "no op" fixup — goes back to same instr or to nxt?
A- goes to same

Q. when does "disable" occur? (A. occurs before branch.)
"enable" occurs (" after branch.)

LASL would like to be able to put in all "no op" fixups — can't be done now.

(Instr. Reads — LASL doesn't like concepts —

IC or IC+1?

question: Two types of interrupts ~~those~~ which (a) happen at end (b) happen during.

Serious → One word fixups — drains look-ahead
— add to memory — can't do here — (underflow — clear to zeros)

([illegible]) Maybe a more conservative view is called for on interrupt

→ { one non-~~[illegible]~~ indexed ~~[illegible]~~ instr.
{ which do not make one lose look-ahead is needed.

Q. Br on bit to addr 0 is there is interrupt? No.
So (I) is known in ~~[illegible]~~ (IC).

(Flag on index reg - Count = 0 + flag is off.
(Sect 5.3, 5.2 changed

Branches do other 'jobs' - modify mem, etc. - you don't ~~[illegible]~~ know where you came from.
(1) ~~[illegible]~~ are these common? size of dist
(2) ~~does~~ one need to know?

Principle: one would like to be able to find instr which did cause interrupt.
in any case without being especially coded for it

for (B) - can branch be prevented?

Interrupts after execution IC is advanced, then int instr in memory
" during execution IC is adv. to next ~~[illegible]~~.
(branch has become a no-op.

→ Try to prevent count from being stepped up if there is an interrupt. Q. successful branch?

would like "data write" interrupt without inhibiting the write instr

Thursday: Feb 20, '58

(6) Space Req.

- would like to know as much as soon as possible

Time lag for govt. bldg. is great enough that it --?

(54 × 54 × 8 room)

(5) I/O:

729's Mod III good shape (60K tape)

×100 tapes: not likely

×10 swift tape: — likely but may not be on LA machine

auto changing? of tapes — not likely.

Printer: 150 l/m is all we can promise now
500 l/m not reliable

asked about Electronic printers, etc.

Plotter: profiles at successive times + read off differences between curves written at diff. times, — a drift of 1%

what are willing to pay for extra precision?

reliability on day to day is important —

(Helsing) (7) auto. Prog.

no detailed plans until past month

Initial Programs:

(1) assembly prog. for 704-709 which would convert Stretch Symbolic to Stretch absolute, — prepare initial testing progs.

(2) Simulator Debugging — execution of internal operations but not all I/O timing. — still questionable — very big job itself

(3) Supervisor Prog. — minimum.

(4) Fortran for Stretch. — no detail thought now

— early date vs. — fancy system.

assembly "709 load & go" system — symbolic mem dumps. corrections.

Language of assembly: — new ideas not complete —

Name	Object
XYZ	Instruction loc. Data area defined
XY	Index register, 2

prog. must keep Track of these,

{ Fl pt. variable, no. of wds.
Fixed pt. variables, bits, Sign, Dec/binary
constants

Examples of Instrs written

IXWD → XZ (load index direct)

IXWD(XY) → XZ " indexed eff. addr.

IXWD → XZ(V) load value

→ XZ(V, C, R)

XZ → IXWD   store index

XZ(V) →

XZ(V) + INCR   (means → XZ(V) understood)

J(V) = CON(I)   a test   ie, "Is J(V) equal to CON(I) ?",

IXR = 0 , PQR → *   Branch   PQR goes to IC

+A   add to accumulator

(objections) — makes machine "too easy" — one never learns of codes

opinion ((LA prefers mnemonic codes . naming of symbols variables too early -))

→ Plea to be able insert hand code sections into auto-code

Important codes are now hand coded —

no, codes being written in Fortran is larger.

→ No. of symbols very important .

(1) Schedule when Things are ready. — very important -

one-to-one assembler + debugger — first a " -

primitive system - same as auto coder uses for inner loops,

Requests LA make a proposal as to what LA wants to do , IBM does , amt of cooperation , what progs. be written.

→ Ed. Voorhees will write letter to Heising by March 19

(Codd) Crude Supervisory Prog.

one prob. program. to do:

(1) Manual supervision

(2) Taking care of "stuck" progs

(3) interrupts not specified

(4) protect itself

Signals: 2 classes

(1) routine eg. loading

(2) not preplanned: eg debugging

OPS: STOP & START

1. Resume PP

2. Release PP all I/O ~~stopped also~~.

3. Stop PP stops CPU, I/O finish, saves interrupts.

ENTER & DISPLAY

Enter & display
instrs
floating data
Fixed data
Data bytes
control wds.

alpha
octal
decimal

certain standard
Radix conv. + code
translation

Load and Dump

instrs
Floating
Fixed
Data
Control Wds

CR
PR
PCH
Tape

alpha
dec
octal
binary w- w/o ECC

Debugging

Clear PP area
Execute
Single Step PP
Search PP (static or dynamic)
Whenever
Leave Whenever Mode

Other:
change source
change destination

ignore
end of item
End of message
Cancel
Backspace

Sense Switches:
Set by & read by machine or operator

→ Max 2000 wds supervisor space is max LA will allow.

~~Same~~ I C

→ (Sayers) Renegotiation meetings will be coming up
IBM's giving xtra mem — a big bargaining point — (32 K, 4 boxes)
1957 year end report was promised — talk into waiting until Jul. — will start then
"horsetrading" at second string level.

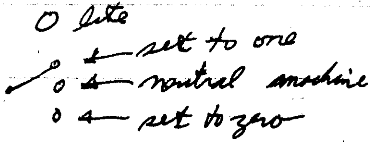
(Sayers) Console:

A. Normal Use of Machine
1. Dynamic s — prob. has on-line output which allows operator to decide on next prob to be done. (interpretive console is O.K.)
2. "change of plan" — wants a buffer to check etc before sending in.

B. Debugging
1. program — "high priority" debugging — real time is the important job.
2. machine — part of engr. console.

Switches + lites. — Two-way — difference Sect II-

— want input — fast interaction not in exchange channel. —



few hundred millisec time --
— second Time scale

— operator must be able to write a 1 & it is read a 1, etc.

— bits in memory are not equiv. to these switches.
bits must be tied to lites

→ Branch set to zero if on these, should be followed by "write console" & set lite. — an administrative decision — not wired in?

(B.1. a small version of what Ted described)

Slowness of typewriter. Display AC, MQ, IC.

Display of arabic digits of cathode ray tubes?

⟹ LA wants 64 switches (32 are felt too small) {really n inputs / not 2^n}

objection to strip switches — (1) typewriter is better
(2) error causing
(3) lot of space

but can change one switch at a time

LA propose

64 sense switches & lites

64 more sense lights (or CRTubes?)

→ register display – important

→ (an output typwriter printing directly from mem in Hex. or octal.
(an emergency use.

(3 words code to do this is O.K. Read, copy etc.)

→ to interupt machine when disabled – a need. (IBM will look into)

Load button

→ would like Punched paper tape involved with typewriters;

– get an infinite buffer, but a time lag between punch & reader.

(IBM has failed to give typewriter to do formula coding –
(LASL will have one made which will use ~~paper tape~~. either off line or on line.

a buffer would be better.

a paper-tape reader adapter

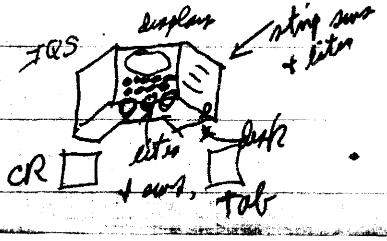
(Brooks) present design.

1. Inquiry station
2. Console
3. Display

} operable by 1 person or 2 people.

propose: 3 Rotary switches 128 points – lightly detented pseudo-potentiometer – no encoding necessary.

console 3 wds (can copy 1, 2, or 3)



a unit signal on each of 3 panels (all same)

Next Meeting.

May 6, 8 — Joint. W. Comp. Conf.

Schedule: Exchange being built

Basic congealing.

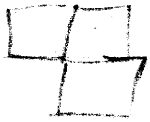
Nov. - still date..

→ { "n boxes of W words"
when can we give final answer,
ask Dunwell.

applied prog.
finance
mkt. est
Mk L line
(sum acc.)

4 way Shifter.

blocks



review I-Box with Foss

(8 xxx) + 11%

- Writing
- 20 eq.
- Test

I/O effect on Mach perf