Stretch comm Meeting Dec 19, 1956

Report on Poughk. Meeting

Agenda of Meeting

1. Significance (anti rounding)
2. Fl. Pt.
3. Data Word Format
4. Addressing
5. Instr. Wd. Format
6. Indexing
7. Selectors

Signs:	S, σ	1 bit
Indicator:	i	1 bit
Exp:		9 bits
Mantissa:		48
"Programmer bits":		4 bits

$\epsilon = (\sigma = i')$ of
$I = (\sigma = + i')$ of
replaces old $\epsilon + I$

add	E	X	e
E	E	E	E
X	E	*	X
e	E	X	e

where * = X or E or e
mant = 0
exp. > l
no. of leading zeros > n

mpy	E	X	e
E	E	E	e
X	E	*	e
e	e	e	e

where * = X', E, e, or exp > l

div	E	x	e
E	E	e	e
x	E	*	e
e	E	E	E

no unnormalized opns. as such

$\frac{0}{0} = \frac{x}{0} = \frac{0x}{00y} = E$

↑ no. of bad zeros.

Optional Breakin on (no unconditional b.i.'s.)

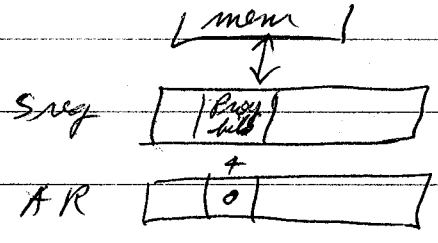
1. E generated
2. e generated
3. improper divisor
4. for E×e
5. for $\frac{E}{E}$ and/or $\frac{e}{e}$
6. for exp > l
7. for exp < -l
8. mantissa zero on add

in masking register
trigger register
summary register (for option declined)

l not necess a power of 2?

Decision: recommend i ≡ z indicator . . 0 = 0, 1 = non zero.

auto. breakin on overflow.



sprog. bits do not go into arith unit

vocab. for handling prog. bits.

	S	σ	(exp)	(mant.)	(prog. bits)	Z
Fl. Pt.	1	1	9	48	4	1
Fixed. Pt.	S	(add, prog. bits?)		48	4	1

Log. Wd. XXXX

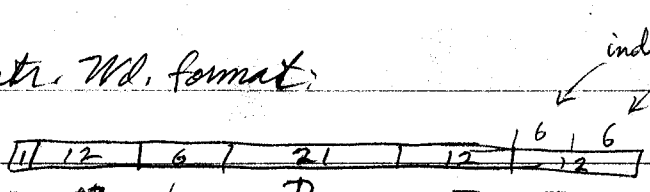
also how two modes: sign or no sign

log. opns how:
1. bit address
2. field length
3. D loc. of wd.
4. receiving reg. starting (bit) address

XXXX

get shifter + C.A.'s ors + mdx?

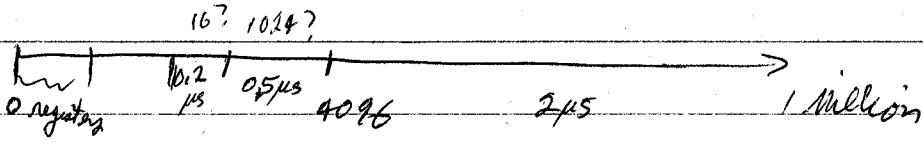
Instr. Wd. format:



the D, T1, T2 can be rearranged for different classes,

---

addressing no. system



00000 ≡ no address C(0) = 0

Indexing:

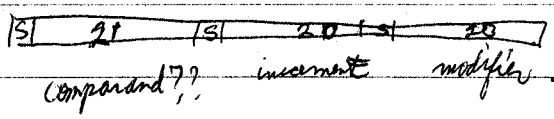
1. direct: $C(D + C(T_1) + C(T_2))$ (calling out no.)
2. Immediate: D and T_1 and T_2 (45 bit no. as fl. pt. no.)
3. Immediate: $D + C(T_1) + C(T_2)$ (21 bit effective address)

question? whether they can do these? two addr.

4. Two address: pre store
5. post store
6. $C(T_2)$ instead of acc [$C(T_2) \rightarrow$ acc, opn, result $\rightarrow C(T_2)$]
7. Indirect addr:
8. Bit address (20+6=26)
9. Mult. tags: tags of 12 bits, as many as 5 — has been dropped?
10. Mult. tags: (geometrical addressing)

(see next page)

Index addr: format



01 20 20 21 ??

f(x,y) base tags
g(y,z) f + x + y

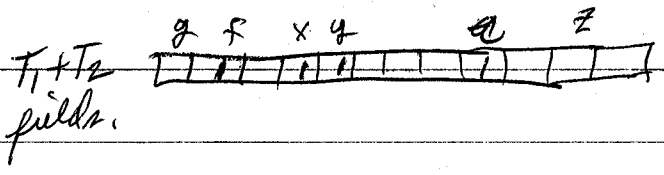
21 bits: each bit corresponds to one of 21 index regs,

Same as 704 except with add not log. add.

and base addr. is in 1st register,

also increments (−1) to give x−1

gives addr f+x+y−1



C(a) = −1 for example

gives block transfers to index regs,

Selectron, (look into!)

A·B ∨ A·$\bar{C}$ ∨ B·C scanning? all at once

more generally by using any 3 registers

A, B, C are any bit address in memory, or any trigger

α	Tr	$\bar{A}$	α+2
α+1	Tr		β → true
α+2	Tr	$\bar{A}$	α+4
α+3	Tr	$\bar{C}$	β → True
α+4	Tr	$\bar{B}$	α+6
α+5			
α+6	Tr	γ	→ false

Test B, b, f

ind addr, bit position (or addr?), field length

any 1 or any 0

all 1's or all 0's

count no. of 1's or 0's

count (when) first 1 or 0 occurs.