

3<sup>rd</sup> Meeting at Los Alamos

Nov 20, '56

Dec.    M    T    W    T    F  
10 - 11 - 12    (13)    14

Next Meeting?

Those present:

Sweeney	Woods	Voorhes
Griffith	Frank	Lozano,
Serahan	Carlson	Blancett
Beckman	Worlton	
	Kolsky	

Topic Floating Point  
~~matrix~~  
Operation format,

In a month:    Data format (cont)    ?  
I/O requirements,    ?

Sweeney: Floating point format

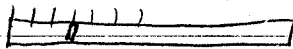
auto sign control  
break in or  
machine generated defs. for expts  
testing  
multiple precision

1 bit exp sign  
12 bit exp.  
1 bit mantissa sign  
48 bit mantissa  
2 bits exception indicators  

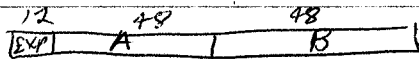
---

64

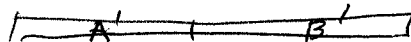
break in feature!



a bit in right position will cause transfer out when what ever exception has occurred, to address given.



with unit (one x.p. for both? or two?)



previous results held until A + B are complete

Sequence:

1. Data  $\rightarrow$  S
2. AB into A'B'
3. ~~recreate~~ result in AB

ie, both operands are held, with self-checking assumed.

Exception Indicators:

I E

I = x.p. overflow E = exp. underflow

These are carried on ~~into~~ if used later if break in is not specified.

if either operand is I the result is I.

E is killed by add opn. (unless both are E)

Load & Store operations

normalize - unnormalize N nM operators.

set pos. set minus M u i ps (use, imat, set pos, set neg)

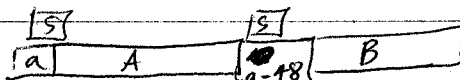
1. Data into S

~~load~~

M load A

M load B

if two is sep. exp for B, otherwise use add.



S<sub>2</sub> The same.

L+S operations:

~~to ~~store~~ A~~

1. M load A

M load B

M load AB

clears B & sets B to 0 correctly.

M Store S

M Store A

M Store B

M Store A'

M Store B'

M (Rd & store A)

Round

clear AB

Store

Add/Subtr opns:

NM add ~~to~~ A to S

imply  $MS + A \rightarrow AB$

B is set to 0 before opn.

NM add S to A

$MA + S \rightarrow AB$

NM add AB to S

$MS + AB \rightarrow AB$

NM add S to AB

$M(AB) + S \rightarrow AB$

NM Sub A from S

$MS - A \rightarrow AB$

NM Sub S from A

NM Sub AB from S

NM Sub S from AB

discussion of normalizing ---

breakin: 1. If exp. overflow

2. Tr to breakin leave acc alone

no breakin: Turn on I, clear AB

If  $x_p$  is in range of  $B \times p$  to  $A \times p - 48$   
 if  $B$  underflows, clear  $B$ , set  $E$  in  $B$ .

Floating Zeros: zero mantissa (both  $A+B$ )

$0 + X = X$   
 $0 \cdot X = 0$   
 $\frac{0}{X} = 0$

definitions are true for  $E$ , what about mantissa?  
 (these will be done for)  
 no breakin.

for breakin: can do whatever we want to, (zero mantissas but valid exponents.)

Note: a different indicator<sup>test</sup> for 0 mantissa than for  $E$ .

Div & Mpy ops:

NM A.S      question of norm, operator.  
 NM S.A

M(AB)/S      quotient A, remainder B      operand 1

MA/S (B default)      (quotient normalized if divisor & dividend are)

MS/A      unnormalized divide? shift dividend right? or prenormalized both?

operand 2

	I	X	E
I	I	I	I
X	I	X	E
E	I	E	E

are also need a null operator?

no breakin,  $\phi + A = A$   
 $\phi \cdot A = \phi$   
 $\phi / A = \phi$

add

	I	X	E
I	I	I	I
X	I	X	X
E	I	X	E

$A/\phi = ?$  what, breakin

addition

	I	X	0	$\phi$	$\phi$	E
I	I	I	I	I	I	I
X	I	X	X	X	X	X
0	I	X	0	0	0	0
$\phi$	I	X	0	$\phi$	$\phi$	$\phi$
$\phi$	I	X	0	$\phi$	$\phi$	E
E	I	X	0	$\phi$	E	E

define:  $X \times p \leq 0 \times p \leq Y \times p$   
 considerably greater

Mpy							Dividend Dir							
	I	X	O	Y	φ	E		I	X	O	Y	φ	E	
I	I	I	I	I	I	I	divisor	I	E	E	E	φ	E	
X	I	X	O	Y	φ	E	}	X	I	X	O	X	φ	E
O	I	O	O	O	φ	E		O	I	(I)	(I)	(I)	(φ)	(E)
Y	I	Y	O	Y	φ	E		Y	I	X	O	Y	φ	E
φ	I	φ	φ	φ	φ	φ		φ	I	(I)	I	I	(I)	I
E	I	E	E	E	φ	E		E	I	I	I	I	φ	(E)

divide by

mpy: I, E 2  
 add I, E, 0 3  
 Dir: {pre (divisor) φ, E, 0  
       Post I, E (same as mpy) 5

10 cases of breaking possible

breaking register



both immediate & summary for each register

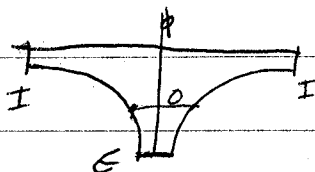
- (leading zero)
1. partial cancellation 5
  2. I' & E' tied to test on "sliding marker" on acc. exp.

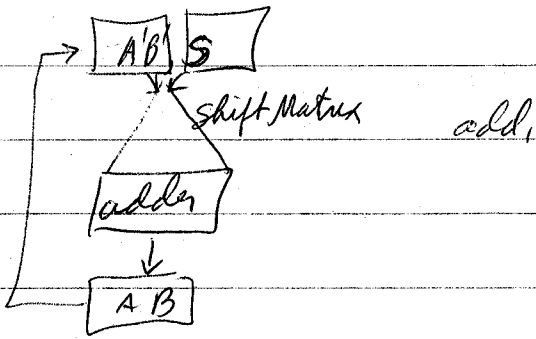
I' & E' are still contained in X in above tables.

Suggestion of Mode where we limit words to fewer no. of bits

e.g. 40 than 48 ? question of difficulty

nos. on reg. scale





$A'B'$  or  $S$  which ever is smaller gets shifted right.

Does this full renormalized work?

question of  $p$  leading zeros? - may be powers of 2  
 " " keeping track of left shifts.

### Special Opas:

1. Sq Rt
2. Round
3. normalize
4. change  $xp$ , to floating pt. no.
5. change mantissa to exponent
6. Borrow ( sub 1 from low order of  $A$ ,  
 Store  $A$   
 insert 1 in high order of  $A$   
 replace  $A$  mantissa by 1 in HO  
 reduce  $xp$  by 4?
7. Interchange  $xp$  &  $A$
8. Interchange mantissa &  $A$
9. Set  $xp$ ,
10. Set mantissa
11. compare  $xp$ .
12. compare mantissas
13. compare fl. pt. no. (assumed normalized)
14. count leading mantissa zeros.
15. add to  $xp$ ,
16. add to mantissa
17. Test indicators, one by one  
 ; either acc. or memory,

Double precision add.

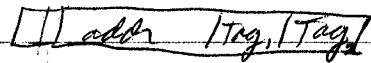
DE + JK

- orders
1. load AB with E
  2. N Add AB D
  3. N add AB J
  4. N add AB K

(takes 5  $\mu$ s for example) for slow mem  
 ~ 2.5 for fast mem

Mpx  
 (DE)(JK)

assuming



1. load A with K
- 2.

double tag  
 "prestore"  $\rightarrow D1 \quad D+t_1 \rightarrow S \rightarrow t_2$   
 "post store"  $\rightarrow 10 \quad D+t_1 \rightarrow S + execute \quad D \rightarrow t_2$   
 " "  $\rightarrow 11 \quad (D+t_1) \rightarrow S, T_2 \rightarrow A'$   
 execute  $A \rightarrow t_2$

reinserted  $\leftarrow$

$D + C(t_1) + C(t_2)$

addr no time  $D + C(t_1) \rightarrow S \rightarrow C(t_2)$

addr 0.2  $D + C(t_1) \rightarrow S \quad Reat A \rightarrow C(t_2)$

addr 0.4  $D + C(t_1) \text{ op } C(t_2) \rightarrow C(t_2) \quad \text{also left in acc.}$

can add in 3 insts using these,

- load X  
 mpy Y  
 m add AB F<sub>2</sub>  
 add A & store F<sub>1</sub>  
 store B F<sub>2</sub>

1. D into S
2. T<sub>2</sub> into A'
3. execute
4. A  $\rightarrow$  T<sub>2</sub>

(Scenery)

Multiple Precision Times

Time. Nov 15, '56

	Single			Double			Triple		
	Insts.	Time	A-time	Insts.	Time	A-time	Insts.	Time	A-time
Add	2	-	1.0	6	5.0	2.6	25 (23)	16.8 (17.4)	11.6 (11.2)
Mpy	2	-	1.6	11	9.1	6.7	37	26.2	22.4
Div	2	-	2.2	10	9.9	7.5			

Fixed Point:



How does hardware work for exp. region for fixed no. transmission of no. particularly questionable.

Question of tags on words. What is rule of combination?

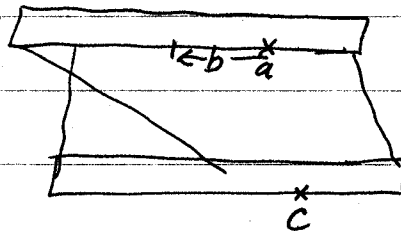
on store is tag stored too or separately, etc.

general store  
address instr:

Radd A  
add (sub)  
Store  
Compare  
Test zero

all have modifiers  
a, b, c.

such that take b bits  
starting at a pos in c



→ Index registers come as ordinary words with faster access.

question of 2 tags:

- can one call more than one fetch with one instr.  
to avoid complete restart to code, new addr.

- can one give instr "add n to previous effective addr.?"

I/O discussion:

considering I/O 8 bit bytes being standard for all



discussion on word length.

S	op	add	add	Tag	Tag
NG		mod,			
1	12	6	21	12	12

data 1 + 8 + 1 + 41 + 5 tags

sign exp. sign mantissa