KMPD P. W. Rocco Blasaur Kolstad

Kingston
Nov 25, '58

added complication - not CPFF machine — can't use present equipment with hope to set up new line
- "off-the-shelf" -

— patent problems, etc would be serious.

what word length?
configuration?
Where is MKT?

Motivation:

- a proposal for New SAGE 36 bits based on RTA-I 5 to 7 times faster, present SAGE not fast enough for all air def. needs. — Bomark is planned other projects for ICBM.
- environment — electronic counter measures, etc. need more speed.
- schedule: present SAGE finished. 37 machines by 1962.
planned: last 7 or 9 machines would be new solid state machines.
Lincoln Lab - wanted 48 bit machine but were willing to take 36
SAC project — 48 bits was proposed. so both

Plans are now that SAC & SAGE II are identical machine
- but what about other customers 433, 488, 46--?
can't propose a new machine for each? — modularity?

Thomas S-2000 was contender for SAC. — lower price.
KMPD machine more flexibility — also had done study of their problems sold our ideas, (They have 704) TRANSAC has slower memory.

$\frac{1}{24} 10^6$ 40,000.
200,000.

SAC delivery 16 mos incl. Testing

"SAGE-SAC" machine 48 bits:

- semi-frozen op. set., different from SAGE & 2090.
- prog. compatibility SAC is anxious to have 704 compatibility.
  will write prog. to simulate new SAC machine on 704. ←
  for preliminary programing + debugging + as standby.
- another prog. to run 704 progs on new SAC.

48 bits, 6 bit bytes, can address any of byte & locate it in accumulator – can't cross word boundries.

SAC needs floating pt., SAGE doesn't.

1	1–9	10–47
sign	characteristic	mantissa
1	9	38

(most signed centered at 128.)

– proposal has been given to SAC & ITT.

Can use this as STD military machine, or design a different machine aimed at future needs? – wide range of customers?

questions:

48 bits may be needed for high speed, but 36 bits can fill wide range.

48 bits 32K mem 200K instr per sec 48 bits aren't utilized as well

- ~~the~~ new MP machine — double 7090 instr speed on adds — added features, etc. runs to 5 times.
where 7090 isn't enough.
(400 - 500 K instrs/sec) — for air defense program mixture.
- bytes handling — gives additional speed advantage, 6 or 7 X
(2½ or 3 X 7090.)

selling price est. crude figures.
without tapes $3 - 5 Million, mfg. costs.

4 SAC machines $26 - 27 M total
(4 computers more $16 M)

I/O questions:

I/O control: control word concept. channels 1 for each type unit.
a frame similar to DSU with pluggable additional channels.
- slower I/O devices — break into computer cycle. character basis.
channels for high speed devices only. eg. tapes
- slow devices — may take 3 refs to memory. to update data.
3 2½ μs mem refs per wd 7½ μs

Tapes: 60 KC per byte, 8 bytes 125 μs/wd. (8 KC per wd.)
could handle 16 Tapes at once.
(actually 2 to 4 tapes are probably max.)

18 bits addressing.

environmental: customers which need.

- not rugged enough ? — how far are we from these now?
- 7090 too slow. Solid state devices are already quite rugged.

eg. BMEWS.

FETA & STRETCH about same speed — same area. —

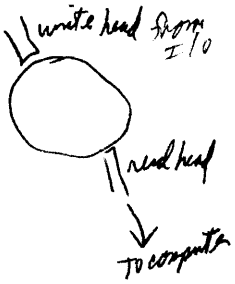
question of overlap of speed & operation, between 7000-X + "SAC"

Should SAC machine be made in commercial plant rather than KMPD ?

Drums: 2½ μs word rate — needs direct channel.
200 K words. (160 K for 48 bits)

question of interrupt:

for SAGE Telephone I/O section (1) a memory buffer
(2) a drum buffer



The computer decides when to read, every 5 secs it checks all inputs. — no interrupts. — completely indep.

on SAC machine: an auxiliary mem. drum only. IT+T will furnish another drum, they will feed an I/O register (48 bits), use programmed breakpoints.

(Nov 18, '58 revised Manual: Fl Pt;, Norm-unnorm. 16 logical connections by Reilly

5 or 6 index regs, initially — can have up to 15.
(18 bits long)
— have considered byte indexing 2^3 bytes in 48
but are not including it now.

I/O 6 bit characters — tape or character tube.
convert from binary to decimal.

several consoles: like I/O devices — switching is done in these frames.
- SAGE type
- maintenance console ← same for SAC,
- duplex console from A to B (don't have these)
- simplex console spares.

Indirect addressing: (Rt. half wd (18 bits) is address).
Immediate adds. (Rt. half wd is data)

---

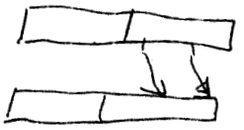
multiprograming? other progs. in memory — have breakpoints to switch between. anticipate progs. & call in before needed from drums.

Weather Prog?
Work statement is due about now — are watching it.
hope to catch it with "Standard Military Computer".
— contractor has time to study before stipulating machine --

---

still have Split accumulator, 2 24 bit additions. (break carry)
multiple decoding
another buffer — configuration control "exchange registers" — for byte work.

-bytes -fixed offsets at bottom of acc.



-split accumulator -given in op. code.

-don't mention prices --. check with Rocco about copy -

Nov 18 '58

Prog. Instr Specs. for SAC Data Processing System
by Dan Reilly

---

2. addressing
   - direct
   - immediate
   - indirect
   - Indexing
   - Registers

3. Configurations
   - Full wd
   - dual wd
   - Bytes

4. Instrs.
   - Load Class.
   - arithmetic
   - floating pt.
   - Store
   - Shift
   - Reset
   - Branch
   - Logical
   - Convert
   - Misc.

48 bit 1's complement on reg
sign 47 or 2 23 + sign

4 bit index
8 bit op code
2^{15} words eventually. (16 bits needed)
18 bits eventually.

Special codes or modifiers for internal registers

	Direct	Indirect
	18 bits	24 bits

multi-level indirect.

index 18 bits (4 expandable to 14) (1st is right accumulator)

addressable registers:

- Test mem
- index reg
- acc
- B-reg, mask
- prog. reg. ?

- logical reg
- clock - real time
- alarm - master, specific, - mask,
- I/O counters (word, address, angular position)

dual

bytes can displace cyclic rotation

Load Class:

op	P	M	A	T	R	B_p	I	I_x	A
5-7	8-10	11-13	14-21	22	23	24	25	26-29	30-47

A = byte activity (an 8 bit mask)
P = displacement no. of byte positions Exch. to right on way to A.
M = mode: 000 - 011 Full Wd,
100 Dual mode,
101 Left mode
110 Right "
111 Twin "
T = sign data tag (change sign)
R = 1 immediate, 0 direct. (restore taken as const. on one)

A = 11011110 P = 010

Byte 0 1 2 3 4 5 6 7
Exchange Reg. X Y

A reg. X Y X Y
0 1 2 3 4 5 6 7

I = indirect address
BP = breakpoint — dtto Transfer of control

Div, clear to zero, on B op. sign

illegal division |div| ≤ |dividend|
?

FL PT.

S	C	F
0	1 8	9 47

C has 128 added.

use same op code format - leaving out: P, M, A, T

1 "Float" instruction.

2 Fl Round FAD, FAM, FSB, FSM, FMP

$\frac{11}{10}$

F Div if mag of fraction is greater than mem

spurious results ! ?

FShift

Unnorm A, S, M (not div)

STORE Class:

Same as load, etc.

"addone to Register" - acc. is changed. (add to mem destroying acc)

EXchge (Swap acc to mem)

Shift class:

OP	C	M	D	....	R	Bp	I	Ix	G	address N
0 7	8-10	11 13	14		23	24	25	26-29	30	40-47

R = real data

C = converting { acc / B, acc only, B - only, Invert }

D = round

G = direction of shift

N = no. of shift cycles.

shift

cycle

normalize

Reset class:

H = half wd. modifier X = indx reg. (J) direct & immediate

Load Index Store Index.

add to Index

Branch Class:

S = ~~Not Branch and Modify~~ Branch and Store IC in an index. (& use in Br) instead of prog. counter (path finder)

Branch relative +

Branch + to address plus Logical Register.

(on bit level – not more than 6)

Br on sign:

A = activity

Left most bit of more than one byte.

Br. on zero. $\begin{matrix}+0\\-0\end{matrix}$ or (± 0)

Br. on logical Compare (contains one byte -- immediate compare)

Br. on Sense Unit.

Br on B plus

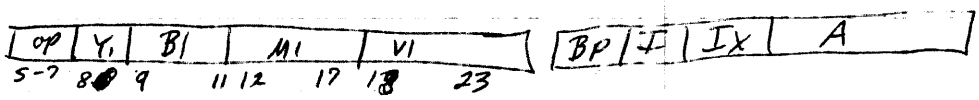
Br on Indx $\begin{cases}\text{high}\\\text{low}\\\text{equal}\end{cases}$ (D = decrement

Br and set Index

Br & Incr. Index

Br on $\begin{cases}\text{pos.}\\\text{neg}\end{cases}$ Index.

Logical class:



V_1 value = inserted into mem - byte

B_1 = Byte to receive info

Y_1 = Mask selector ($\begin{matrix}0 = \text{immediate } 12\text{-}17 \text{ of } M_1\\1 = \text{logical reg.}\end{matrix}$)

L_2 = First bit

L_3 = no of bits.

R = $\begin{cases}\text{immediate}\\\text{direct}\end{cases}$ auxilliary logical reg.

set sense of accumulator : (sign [redacted] - complement)

convert :

(16 cases)

Load + shift.

store + shift. (sort of table lookup)

Convert class : similar to 70 ?.

replace A by B reg.

add for b reg.

Misc :

Halt

Store

no op

CAS

C logical

Execute