Discussions re S/360 version of ACS vs ACS

[54/G]

March 18, 1968

G. Amdahl, J. Earle

position – unusual approach

study of circ. counts : 5 level cycle — 8 ns machine cycle.

OS 360 code performance.

(NSS cycle has also been change)

In theory as well as in practice S/360 arch is faster

ACS 7 level
past 12.5 ns cycle

ACS can be done either:

FAT (CPU 265K ckts)	or LEAN (CPU ~180K ckts)	AEC 360 LEAN (CPU 90K)
7 levels present design	6 levels are possible?	5 levels
[3 X The circuitry size and cable lengths also contribute to cycle (FD machine not	(1) binary preshift + binary postnorm. limit them to 6	1 X The
	(2) too many functions	
	(3) too many registers (30% more) fan-out to select fan-out to fn. units	

physical size goes ~ $\sqrt{\text{no. ckts}}$

~ $\sqrt{3} \approx 1.73$x longer wires in ACS

$\frac{1}{5}$ of 8ns is in wire length

$\frac{3}{5} = 1.6$

$\frac{1.6}{3} = .533$

$.533 \times 1.732 = .866$

.546
.9206
1.067
1.987
×7

Note ⟹ 13.909 ns expected

not 12.5 ns

the key problem

cost is 70% in perf. +40% hardware (5 to 6 cycles)

cost is 40%

56% = 12.5 / 8 cycle

what does architecture buy back? maybe 10%

performance / cost ratio is 4:1

E. Bloch a resource-allocation question for SDD

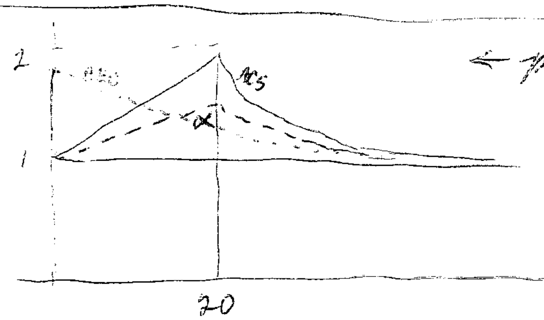
hardware solns to branch + skip are better than ~~software solns~~

for short loops or "tree" code. can fetch ahead about 2 branches

~6 instrs ahead & no "EXIT" ...

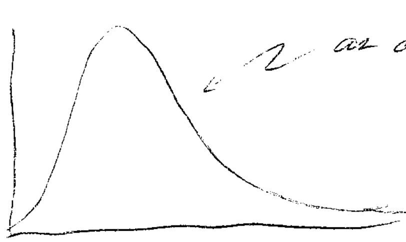
Case
eg 3500 ns for 180 instr or 360 fewer instrs
~ 9000 ns for 240 ACS
[EXITS + lack of RX raise the total

ACS can't branch ahead more than one



ACS 160 MIPS means
~ 2 instr / cycle average

α is



32 bit is faster (than 48-64)
cost of hardware
clock cycle

Scientific Job Shop
measurements of M Drummond.
32 bits 70% utilization 80%
64 30% 20
(ave ~41 bits) Fernbach says

More in memory gives big saving
in speed
I/O

Dec to binary conversion not in ACS ~5% of time is spent in this in assembly, prog counts, address } + compile
other instrs are less.

{ one 9/ order was
{ lost here

can ACS really go to 6 cycle?
engrs say they can't.

2 possible new programming systems.
ACC can use old OS but not limited to it.

Architecture ACC + ACS is main decision
ACS spent their money on exotic coms

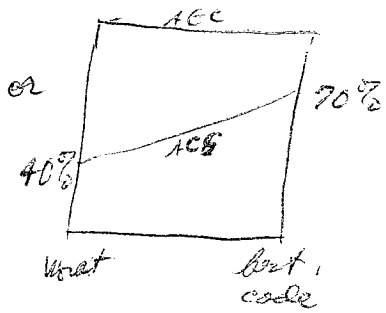
Channel arguments more specifics.
not limited to using Mod 91
Bits of I/O ~1 bit per instr executed

- Bandwidth argument

Example kernels: 412.5 ns to do a loop
288
124.5

$\frac{124.5}{288}$ = 43.2% faster
would predict 56.2% faster just on cycle time
no ACS architecture buys ~9%
three address with buys ~5%
more registers buys ~5%

ACC 2.5X better than ACS on dirty code
1.4X better on best case



~~Cycle time~~

2 Theory ACS lean can do no better than 90% of 360

~~In~~ practice it can be much

~~because~~

to do better they would have to throw out

(1) prepare to branch

(2) 48 bit word

(3) extra registers.

(4) binary fl. point

put in RX etc. Then what is the difference ??

from 360 after all these changes.

Any actual mix of pure execution problem

AEC 43% faster

1/3 of adds were double in the sample

{ 32
{ RX exponent ← a key time diff.

Execution time AEC 43% faster (rather than 56%)

so architecture was only ~9% gain

OS: (1) can run with change in OS as is

(2) improved OS? 2 categories

(a) optimizing compiler -- much is machine indep., -- hardware branch optimiz.

compiler must do extra register allocation on ACS

(b) "nucleus concept" for OS -- can work on 360 as well.

also machine indep.

none of the above depends on 2 instr counters, etc...

Peak execution capability of AEC is higher than ACS, but neither have meaning

ACS 480 MIPS

AEC 500 MIPS

In principle ACS is small step backward
practice large " "
In terms of MIT is an enormous step backward

Base Registers? (1) not apparent problem in executed code
(2) doesn't make up
(3) can be eliminated if necessary

"deep by pass" is good hardware

of 19 SLT boards in Mod 91 15 are in [illegible]

ACS: 12 bit [illegible] 48 bit word
AEC: 8 bit 32 word

ACS register alone is 50%

(1) ACS

only if This is dead and line couldn't get money back) not reasonable argument

(2) ACS + 360

acs does not have enough to offer to make up for its loss.

(3) 360 only ?

direct extension of product line...

(This machine can be US as easily as 360)

no advantage to other architecture unless there is a

What will be the "measuring stick" for machines in the '70's

Summary: In Principle: AEC360 is { Perf. 1.1 × ACS
Cost 0.7 × ACS

In practice as is 360 is { Perf 1.4 to 3 × ACS
Cost 0.35 × ACS

In Market Place:

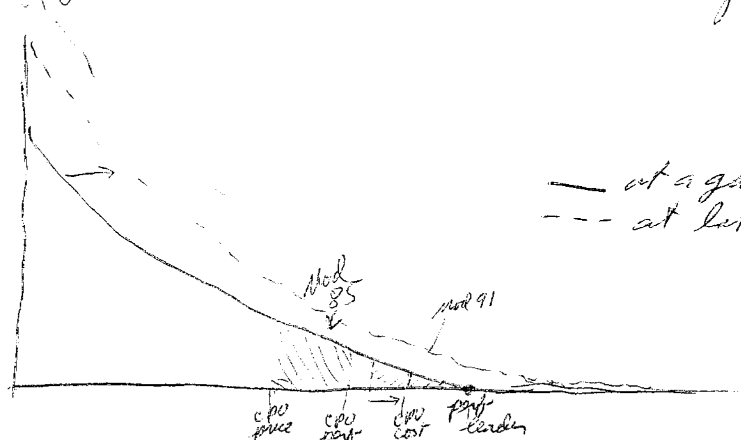
AEC 360 ~ 20 × Mod 85 {3× from the technology alone}

2 jumps ahead of anyone have accomplished in the past ...

AEC 360 can go — ACS can't
can fulfill
can be profitable

actual job mixes ~ 2:1 in favor of 360

Market Place:



CPO ≈ $\frac{1}{4}$ system price

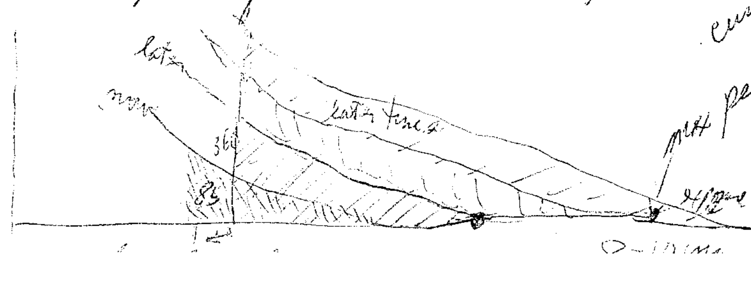
In devel cost & tooling costs. [illegible] increases cost & limits the mkt. "cost spiral"

2 limits of "band" lower: CPO price
upper CPO perf.

but now one can't afford sufficient I/O [illegible] and memory to support max performance ← The CPU is not the limit now

AEC 360 is in cost ball park of 85. can buy ...

customer can pay — max perf units needed } up to factor of 4 the max perf.



{N55 is very narrow in space and time

max perf with Mem & I/O
we are safe up to this point

Dominies P.D. ... [illegible] ... twice the life expectancy

32 ~ 64 - what about 128 ?

5 levels 8 nt

70% 32 pt 30% 64 pt ?

80 20%

50 50 ?

April 2, 1968

G. Amdahl

Q: Relative speeds on Floating Point Problems?

A. [illegible] 32 bit operands — 1.00

" 64 — −15% speed degradation

" 128 (ca '85) — [illegible] { −50% speed degradation in adder; −75% for mpy }

If one has a mixture eg. 80% 64 bit 20% 32 bit

The degradation will be 0 to −5% degradation on 64