Feb 25, '59

Polivka: Console Program

1. Do simple things simply.
2. put data in in "natural" language
3. don't tie up all switches.
4. tied in with supervisor

want: address stopping
dumping control
enter data
display data
branch trace
x

address mode – where from where to – data
format "
function "
syntactical "
when

"when" mode

Scalzi: Interruption Table

goto "Failsafe" – subroutine for any interrupts not specified.

pseudo op. "replace interrupt table by"

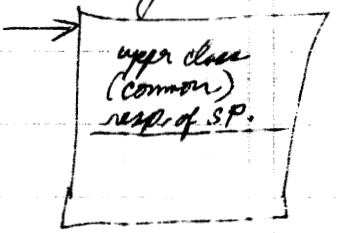
– one instr fix up – is possible provided not "bx disable" – nor "failsafe".

insert bx to SP

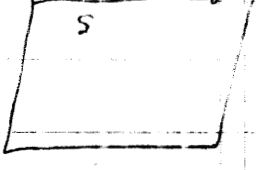
also SIC/ is forbidden – could go anywhere.

Primary Interruption Table

addressed by interrupt addr Reg.



Secondary interruption table



and middle class prog.

done by loader?

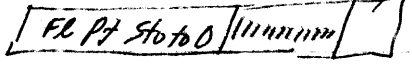
Pseudo Op : System

Entire System
~ 20 K instrs.
Can't what I'm in ?

I/O : Read
Read with Retrial
write
write with Retrial
Rewind
Space block
Backspace block
Space file
Backspace file
Write Tape mark

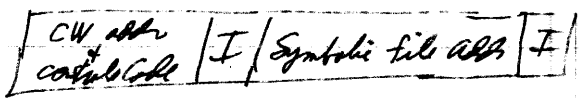
Other : Replace interrupt Table entry by
branch disable / SICIBD
branch enable / SICIBE
Wait
Wait any
set clock
End
Go to Failsafe

Prelude of Pseudo op

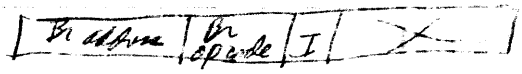


Pseudo op code.
user invalid address interrupt

I/O



pass PP → BE/BD



SICIBE/SICIBD
Replace interrupt table entry by
set clock

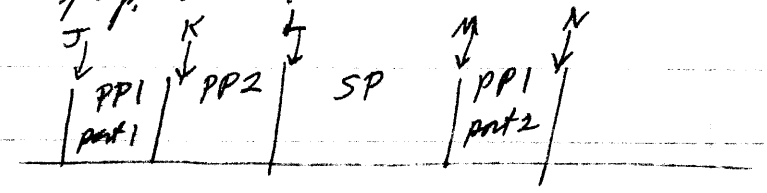
data fetch } must always be 1 — will be cleared occasionally by SP.
instr. fetch } — could be bad if

1. reliable } classes of programs { don't check I/O control words.
2. unreliable { check but don't move.
3. very unreliable { move control words to SP.

for speed prob [data] [data] IF masked off. — reliable progs only.

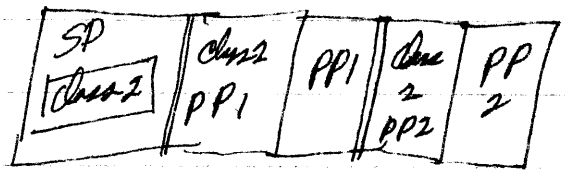
if very unreliable progs are next to each other — 1 block to unmask IF

Example



when PP1 is in LB = K , UB = M
PP2 K , UB = L

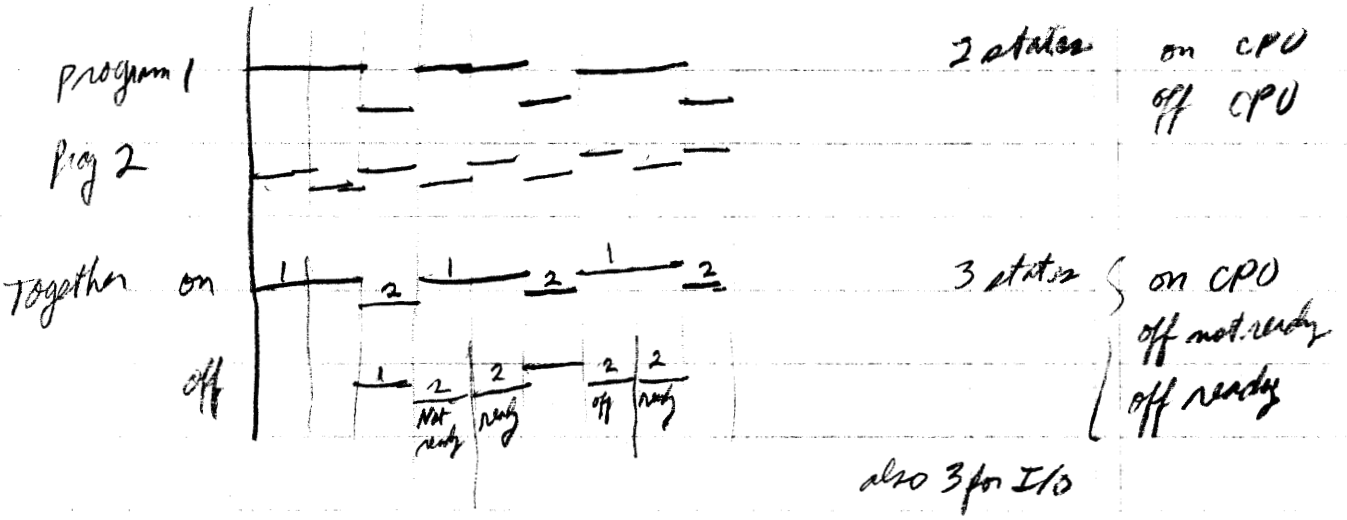
probably only 1 separated prog at a time can be in at once.



I/O Symbolic actual Table
Status Call / I-O Hold Table
File address Call
I-O Pending / I-O active Tables.

Note: "locate" is not used by PP.
"CCW" " " " " ".

Pending
Active
Hold



Rule: don't displace prog. on CPU,

but this isn't a satisfactory rule, because Prog 1 might require least amt of Time compared to Prog 2.

Whenever Prog. goes off machine accumulate Time in ready status + Time in not-ready,

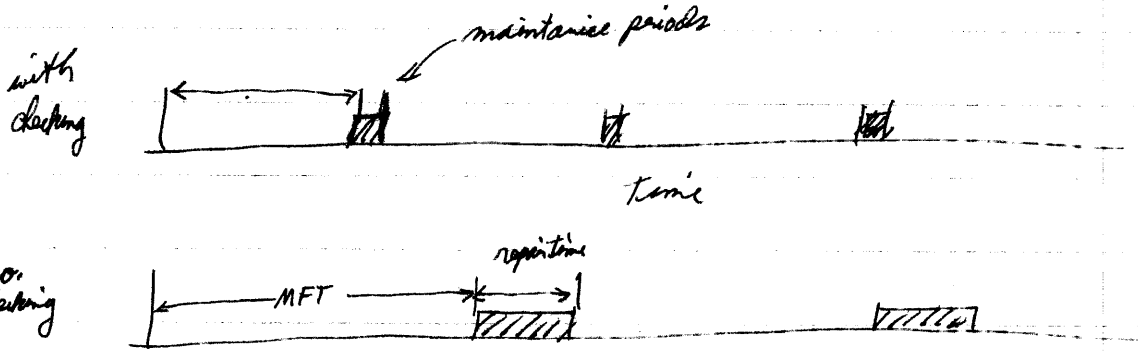
(1) Take longest Time or (3) random choice
(2) Take longest ready time

Frequency with which prog. goes on + off — give priority to higher "oscillations"

compare (Freq, expected length of use)

Stringfellow:

Purpose: To reduce field maint. time rather than for customer's requirements

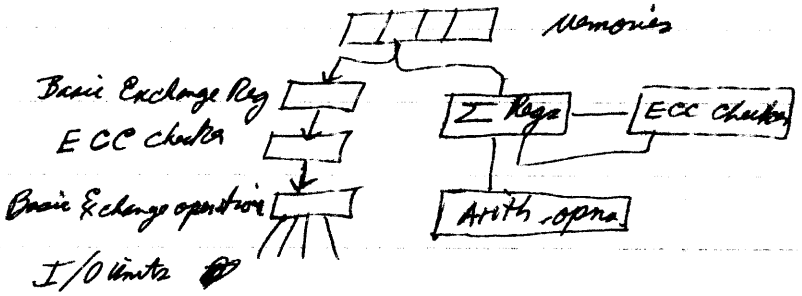


MFT mean time to failure

putting in checking hardware decreases MFT but reduces length of repair time

question of "balance" in system. raises over-all time of good performance.

- convenience: — repairs can be made at more regular times.



1. don't know how to carry ECC code thru Arith. opns. — discontinuity here.
2. customer may select ECC mode or not on I/O units.

arithmetic operation checking:

1. parity prediction: (is used in adding unit, counter)

$$A + B = C$$

use $P_A \neq P_B \neq P_{carries} = P_C$

carries: 0 1 1 1

A: 1 0 1 0

B: 1 0 1 0

C: 1 0 0 0

$P_A = 1 \quad P_B = 0 \quad P_C = 0 \quad P_{carry} = 1$

$[1 \neq 0] \neq 1 = 0 = P_C$

but this is time consuming

for mpy or div must be done for each stage.

2. "Casting-out" process: [can cast out anything which is divisor of 1-base]

Sigma use 3, "residue".

In decimal: 9, 3

In binary: 3, 7, 15 (5, 3)

$$A + B = C$$

Then $R_A + R_C = R_C$

example:

$A = 25$ — casting out 3's: $2+5=7$, $R=1$

$B = 17$ — $1+7=8$, $R=2$; $R_{sum} = 0$

$C = 42$ — $4+2=6$, $R=0$

Also for multiply: $A \cdot B = C$

$$R_A \cdot R_C = R_C$$

3. Hamming Code: — —

- Triple error looks same as single
- errors in checking hardware itself are fixed

use of detected information: when error occurs,

"Scanner":

1. Stop Signal in slightly less than one op time on ECC, record 2400 indicators on I/O device (perhaps on 3 cards? ~~4 cards~~) (from registers, triggers, etc.) in 8-bit bytes. -
2. Exchange: - latch circuits - freezes data & records it at its leisure.
   - could use Exchange.
   - - 6 to 8 hours per week -

- a console on exchange & CPU separately,
- Engineering Changes - RPQ — will be not be as frequent - ?
- changes in field - by whole panels wired in factory,

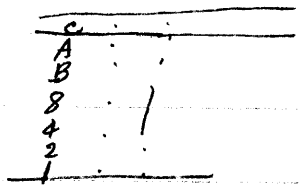
Control checking: only some — data checking does a lot of it.
~~"~~ "data itself up" — too much hardware. -

Marginal Checking: — will be in.

Controversial subject; degradation
- can transistors be predicted? — calculations show it should work
  - larger currents are required in power supply — so it is costly,

- printed circuits — wire wrap. —
- card contacts - gold plated - good for so many insertions - thousands -
- hairline cracks
- no. of Trs: Σ = 160K, 250K for whole system } 25% checking - fault loc.

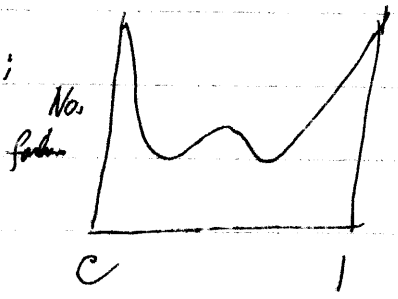
Tape Drives & Format:



} present low-density drives { 727
729 I+II

On analyzing errors on tape:
- wrinkling
- skew
- foreign matter.

{ location of failures; on tape.



would like to put less used info. on edges. — this can be ignored for low-density drives.

by ~~changing~~ moving	7070	729 III+IV.
bits around - get 50%	7090	
higher reliability in high density.	7030	

old: C B A 8 4 2 1 low density 2.5 to 2.3 } for BCD not binary
new: B 8 A C 2 1 4 high density
letter

→ rough on dual density tape.

$50K to convert all of drives
½ day to convert each drive

?
500 1 bit errors.
300 detected not corrected.