

# PROCESS READ ONLY

P.WRITEN

P.ROHEAD

P.EVENT

P.INTR

P.SCHED

P.SCR2 = P.ROHEAD

P.XPACK

P.CLIST

P.CTABLE

P.STACK

P.SUBPDT

P.MAPESM

P.MAPSM

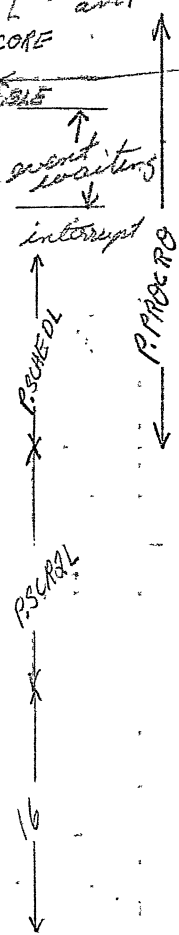
P.MAPAT

P.OLDP

P.INTERR

P.IPHBT

# CH WDS(1)	PROC MET	PROC LENGTH IN CORE
RA+1 INTERRUPT	C-TAB REL	LEN of PROC VARIABLE
event datum		PROC MET
INTERRUPT DATUM	INDEX OF SUBP	
3 clocks 1) user time 2) system time 3) swaptime		
PTR TO HEAD OF FULL CLIST		
LEN/2 of LOC C-BUF	LEN/2 of C-TABLE	PTR TO ORIG OF C-TABLE
STACK END PTR	STACK ORIG PTR	STACK POINTER
LIST SUBP + 9	CRIG of SUBP TABLE	# of SUBP
ORIG of COMP MAP	LEN of COMP MAP	LEN of MAP/ESM
PTR TO HEAD of MAPS IN LIST		
LONGEST TIME		
flag for CALL		
interrupt pending		
input parameter list addr		



incl. scratch and param and user code

{ c-list table THRU compiled maps

- P = something pending ⇒ check other bits
- W = wake up waiting
- R = running
- I = process interrupt to classify
- D = destroy
- E = "0" for ECS process - "1" for pseudo process
- C = process in core
- V = event waiting

- running states
- W=0 R=0 ⇒ blocked
  - W=1 R=0 ⇒ blocked w/ event in
  - W=0 R=1 ⇒ running
  - W=1 R=1 ⇒ wake up while hanging on EC

Line "LEN/2 of c-table"

SUB PROCESS DESCRIPTOR

addressability prev. descriptor length  
 → class cm code = B1+RA

logical map entry

UNIQUE NAME	MOT
FILE ADDR	
C/M ADDR	WP COUNT

0	RA+FL R1	RA R2	BACK POINTER R2
1	CLASS CODE		
2	ENTRY POINT -0-	MAP ORIGIN R2	C-LIST ORIGIN R2
3	# of subp for compiled map	MAP POINTER R2	COMPILED MAP PTR
4	INTERRUPT DATA	MAP LENGTH	C-LIST LENGTH
5	C-LIST UNIQUE NAME		C-LIST MOT
6	BEGINNING of ESM	MAX # of ERRORS	MAX stack ptr
7	-0-	-0-	MAP LIST CHAIN

C-list

length	← MOT Pointer
OPTIONS NAME	TYPE MOT
OPTIONS NAME	TYPE MOT
OPTIONS NAME	TYPE MOT
OPTIONS NAME	TYPE MOT

option bits

STACK LENGTH - (# of ANCESTERS \* 2)

ESM

error class 0 → 31	/
" " 31 → 63	/

CALL STACK ENTRY

forced ESM

OP-COUNT	TOP of PATH	SUBPROCESS
	IP LIST ADDR (REL)	R-COUNTER

class code capability

options 1-15	class code
4 2 15	

Full C-list table

unique name	length	← current sub-process
/	MOT	
unique name	length	
/	MOT	
unique name	length	
-0-	/	← c-table

subp data

last subp + S	orig of 1st subp	# of subp
---------------	------------------	-----------

stack c-list

stack length ← # of stack ptr	stack orig.	stack pointer
length of local c-list buffer	length of c-table	pointer to c-table

c-table

subprocess memory

-0-	RA
-0-	RA+1
type of call	RA+2
address space orig	RA+3
C-list orig	
map orig	
parameters	

call = 1  
 jump = 2  
 error = 3  
 interrupt = 4

stack ptr

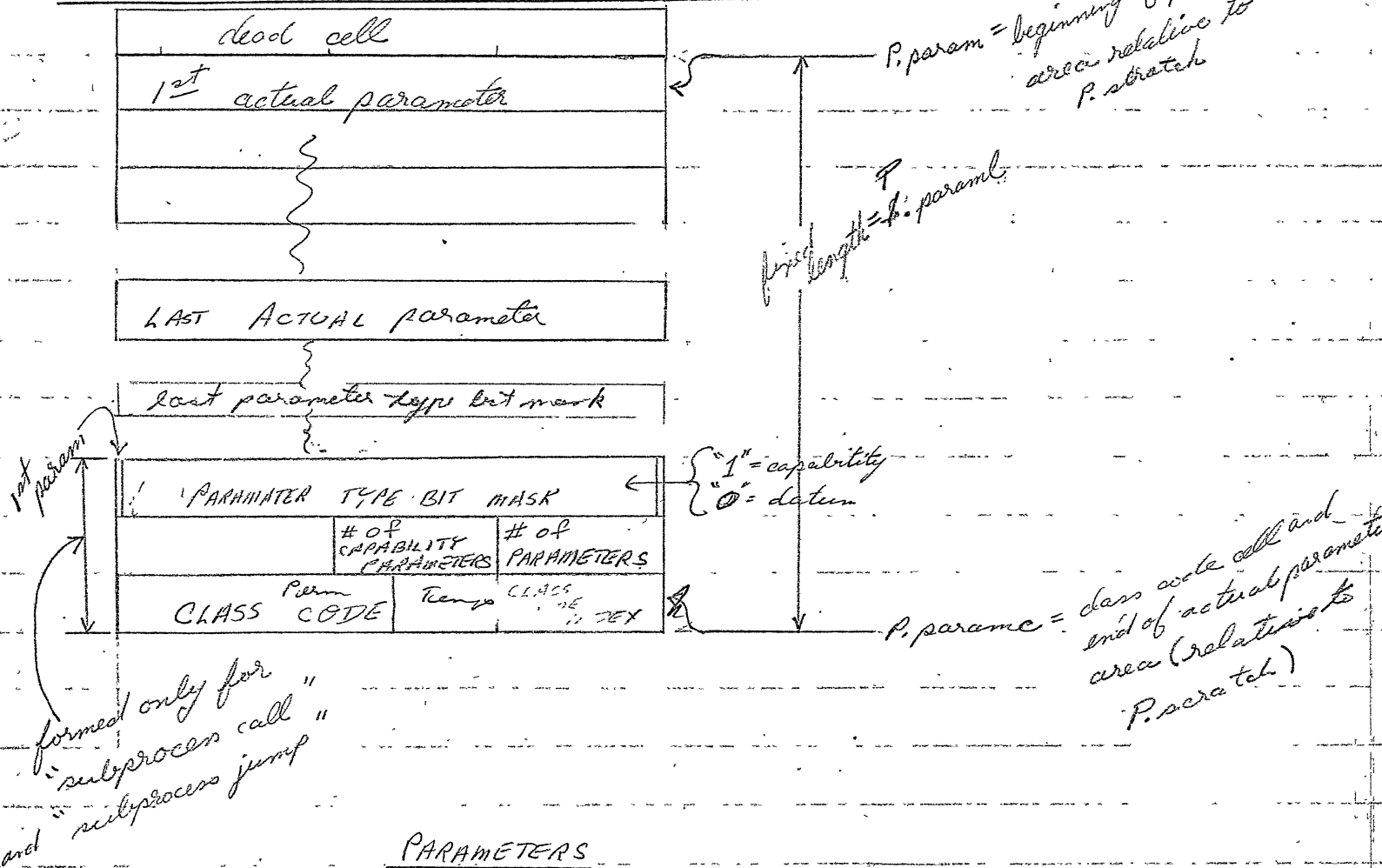
length	← 1
	← 2
	← 3
	← 4
	← 5
	← 6
	← 7
	← 8
	← 9
	← 10
	← 11
	← 12
	← 13
	← 14
	← 15
	← 16
	← 17
	← 18
	← 19
	← 20
	← 21
	← 22
	← 23
	← 24
	← 25
	← 26
	← 27
	← 28
	← 29
	← 30
	← 31

stack pointer

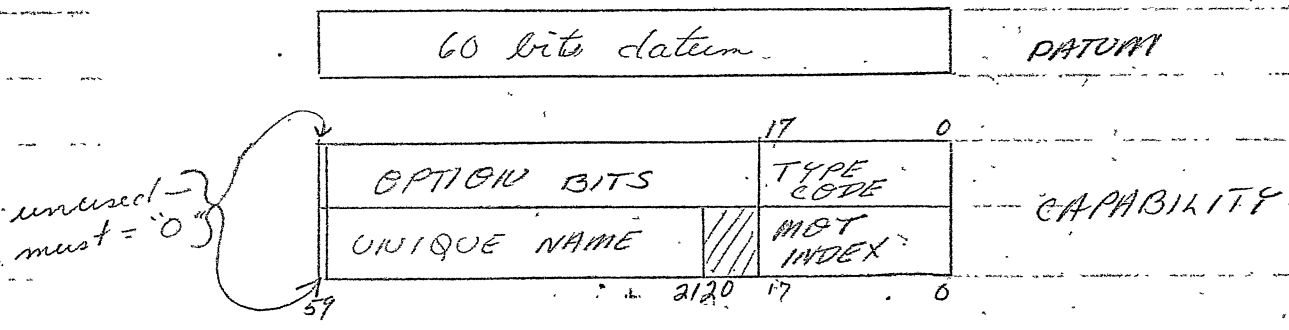
TOP OF STACK

stacking

# ACTUAL PARAMETER AREA



## PARAMETERS

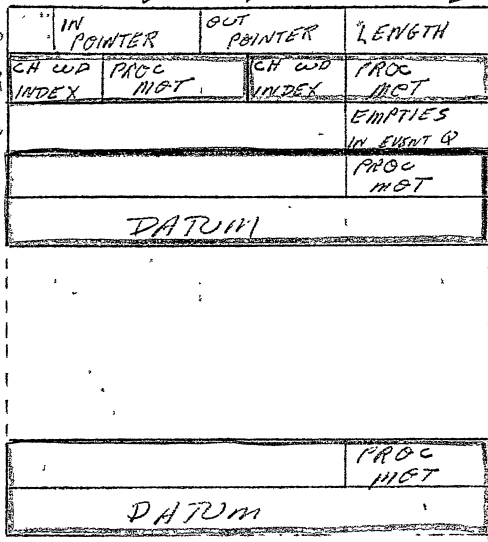


# EVENT CHANNEL

run from 3 to (length)-1?

# evts in whole event channel

pointer to head of process queue

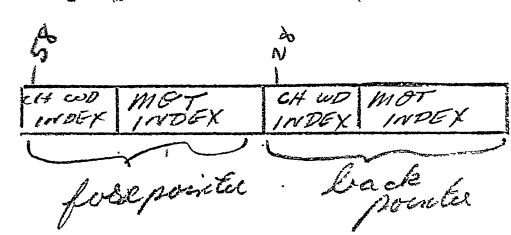


pointer to tail of process queue

header

event queue

- in pointer points to next place to put event
- out pointer points to next event out
- empty process queue will have pointer to self in EC w/ bits 58 and 28 set



contradicts

CH WD INDEX based to work with unpack inst.

Bit 28 or 58 indicates pointer is to event channel.

fore-pointer only?

responses to "send event"

- 1 = event queued (EC.QUEUE)
- 2 = event passed to process (EC.PASS)
- 3 = gone lose (EC.LOSE)
- 4 = full queue (EC.FULL)
- 5 = duplicate event (EC.DUP)

Clocks

(S.INTERV) ~~(S.INTERV)~~

2 clocks - interval timer - counts up until  $\geq 0$

(S.MASTER) master clock (if ~~S.INTERV = 0~~ then ~~S.INTERV~~ not incremented)

(S.CHARGE)  $\rightarrow$  doesn't run during interrupt  
PPU steps both -

if interval clock becomes pos. it was neg  
end of quantum signaled (see <sup>1/22</sup> Timer)

Central maintains elapsed time in the following

categories -

(S.OLDTIM)

1)  $X \leftarrow S.MASTER$   
2)  $TIME \leftarrow TIME + (X - S.OLDTIM)$   
3)  $S.OLDTIM \leftarrow X$

~~me) look in interrupt  
to get elapsed time  
1)  $X \leftarrow S.MASTER$   
2)  $TIME \leftarrow TIME + (X - S.OLDTIM)$   
3)  $S.OLDTIM \leftarrow X$   
4) ~~...~~  
5) ~~...~~~~

system  
user  
swap

use to  
charge  
user  
grand  
totals into  
IPU

PPU interrupt code - on entry

$SAVE \leftarrow S.MASTER$

at swapout  
update clocks last of all

~~leave  
1)  $X \leftarrow S.MASTER$   
2)  $S.OLDTIM \leftarrow S.OLDTIM + (X - SAVE)$   
3)  $INTERRUPT \leftarrow int\ timer + (X - SAVE)$~~

process in ECS

