

22 July '71

I've made the suggestion about a month ago.

A (tentatively) rejected it because the bits shouldn't be assigned to trivial parts of the process R/O descriptors, ie 1 bit should suffice for the FTSE flag he suggested.

The 23 bit in ECS RA is a funny one.

10

VC
Re:

ECS global registers & protection

The global registers work in following way:

bits in X0 are compared with special registers in ECS controller. Escaped by having bit 23 set in X0 and in ECS RA.]

(among others) following things occur:

ACTION = Read/Select

If the set bits in X0 are clear in g. registers, the bits are set in the g. registers and the instruction does a normal ECS read/write return. Otherwise it does an error return

eg	bits in g. registers	... 1100 ..	
	bits in X0	0011 ..	
	result in g. registers	1111	+ normal return

	bits in g. registers	1100	
	bits in X0	0110	
	result in g. registers	1100	+ error return

SAD BDECH

BXD XI

RECS 1

JP *

Batch count the bits
tested + an "action"
number decoded by global
register hardware

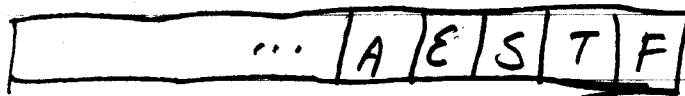
also of interest: selective clear

g. reg = 1100

XO = 101

g. reg (result) = 1000

suggested use



F = P. ahead

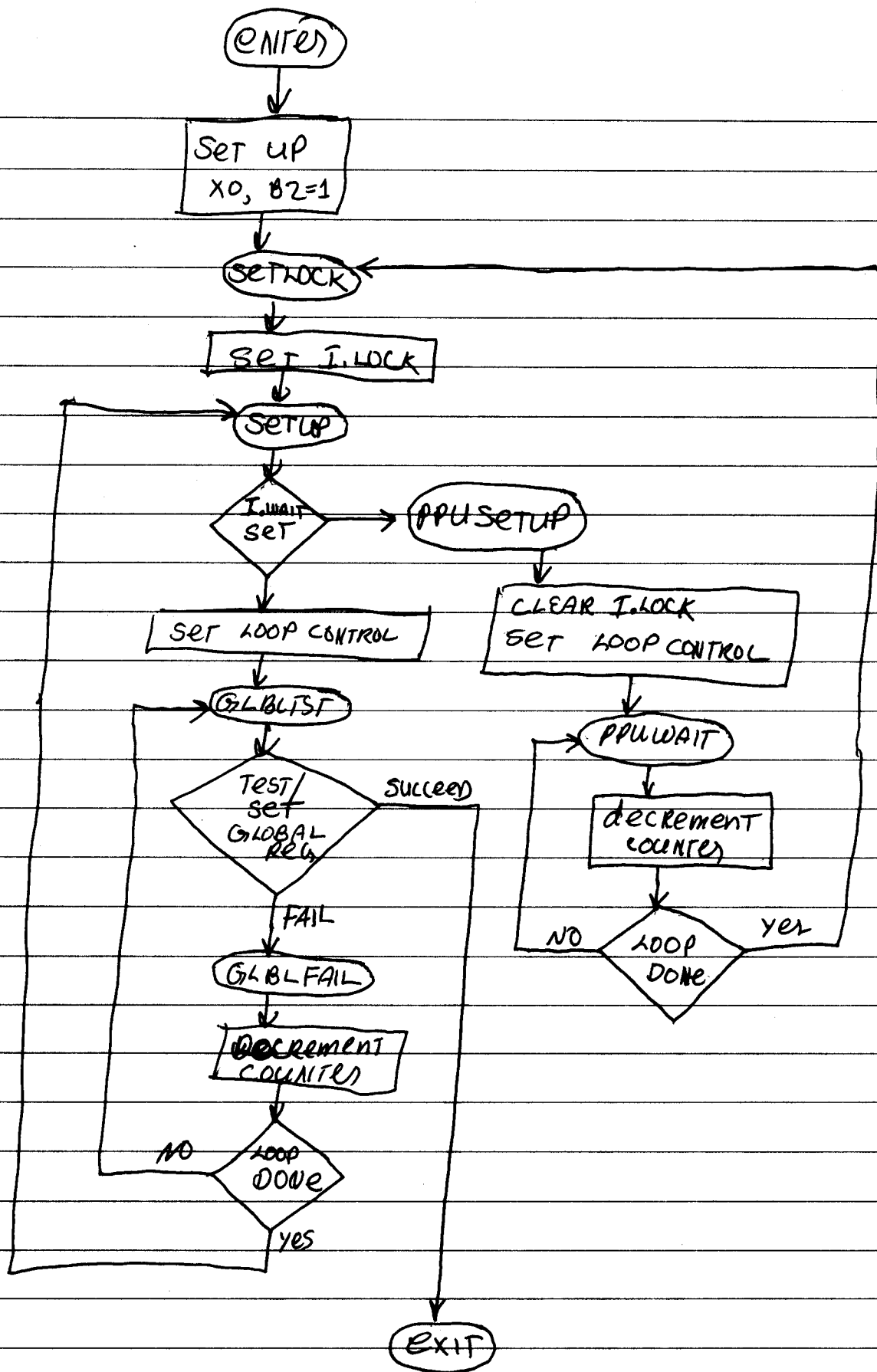
T = process time

S = schedule data

E = event channel queuing words

b = allocation block

anyone who wants to touch a
"protected" datum or object of
above type must successfully do
an appropriate "ready select"
before it starts and a "selective clear"
when it is done.



* GLOBALCHK

* MACRO which loops until it succeeds in
* a test-and-set action on the ~~global~~ global
* register

* GLOBALCHK where : where is address of
* a word which will be placed in x0

* user X0, A1, X1, A6, X6, B2, B3

GLOBALCHK	MACRO	WHERE
	LOCAL	SETLOCK, SETUP, GLOBALTST, GLOBALFAIL
	LOCAL	PPUSETUP, PPUWAIT, FIN
	S A 1	WHERE
	B X 0	X 1
	S B 2	1
SETLOCK	S X 6	1
	S A 6	I LOCK
SETUP	S A 1	I WAIT
	N Z	X 1, PPUSETUP
	S B 3	GLOBALLOOP
GLOBALTST	GLOBALTST	GLOBALTST
GLOBALTST	R E	1
	J P	GLOBALFAIL
	J P	FIN
GLOBALFAIL	S B 3	B 3 - B 2
	N E	B 3, B 0, GLOBALTST
	J P	SETUP
PPUSETUP	S X 6	0
	S A 6	I LOCK
	S B 3	PPULOOP

PPUWAIT

SB3

B3-B2

NE

B3, B0, PPUWAIT

JP

SETLOCK

FIN

BSS

0

ENDM