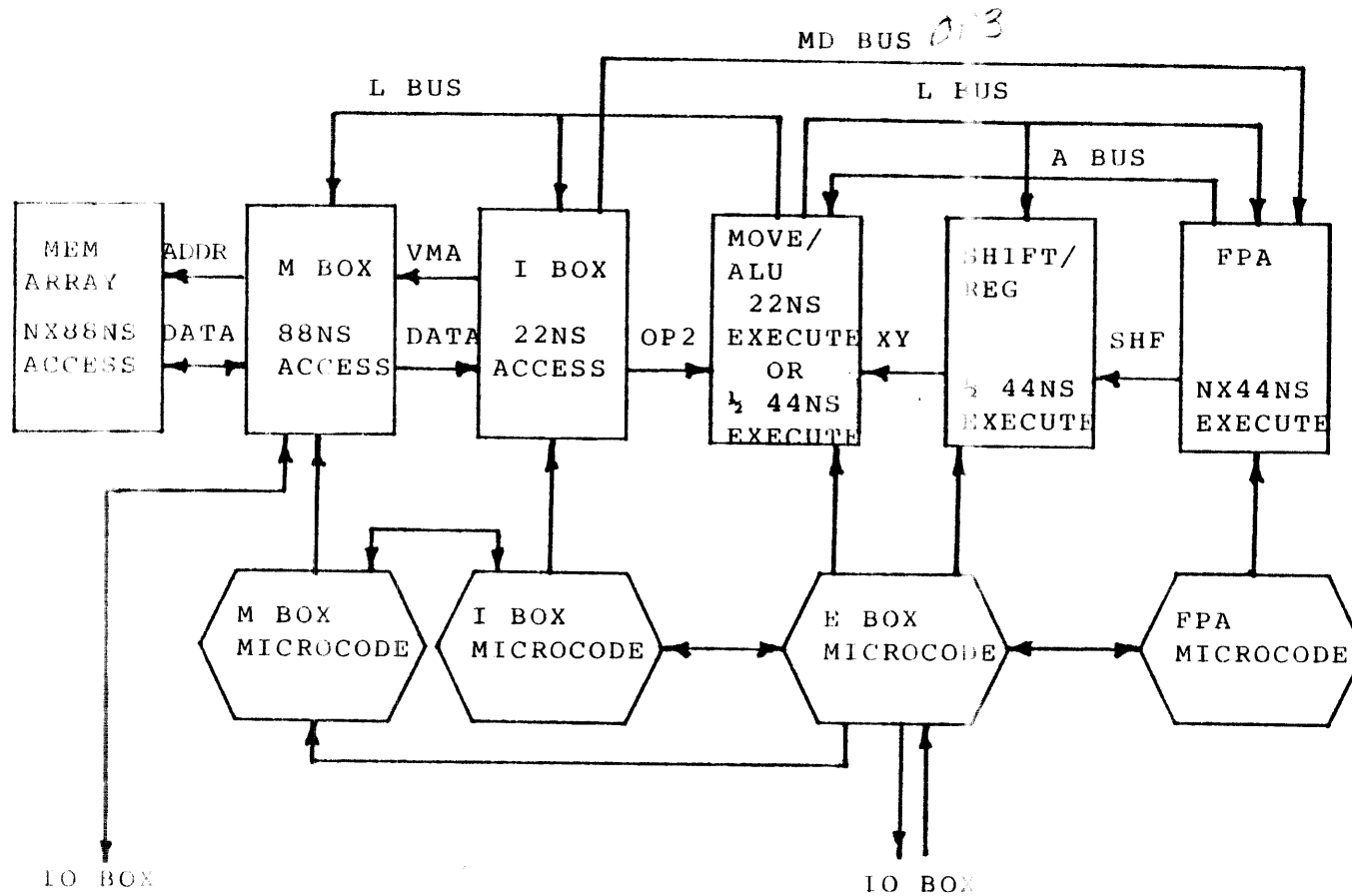


PROJECT JUPITER  
2ND LVLS  
D. HOOPER  
18 AUG 80

DEC COMPANY CONFIDENTIAL

GENERAL

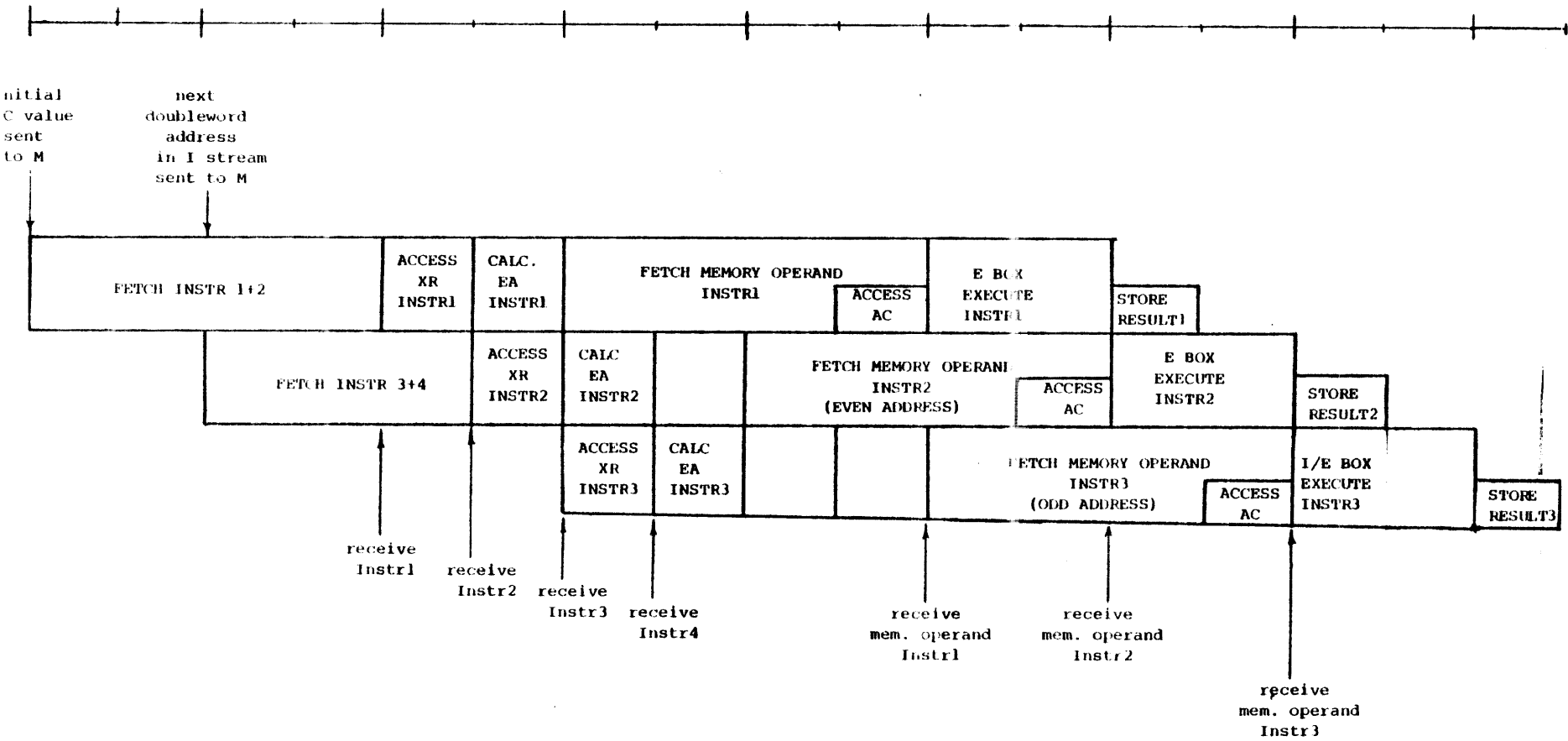


KC10/  
2080 CPU  
BLOCK FLOW

DEC COMPANY CONFIDENTIAL

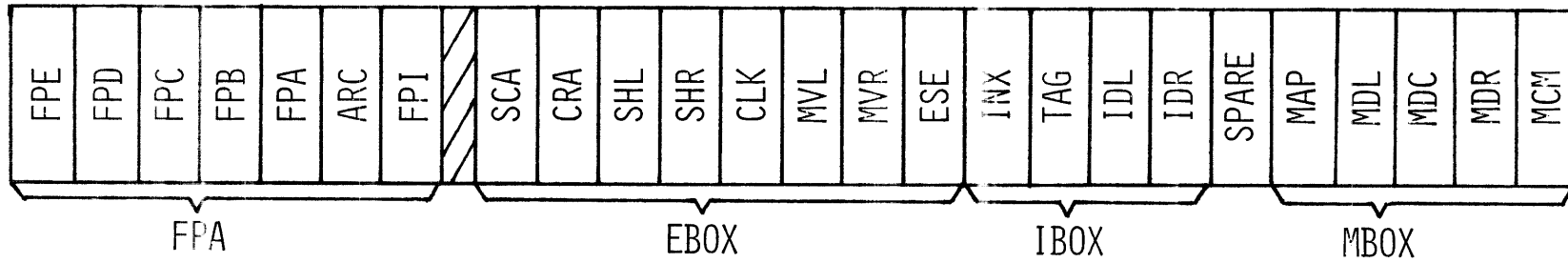
# KC10 /2080 CPU PIPELINE

22ns



DEC COMPANY CONFIDENTIAL

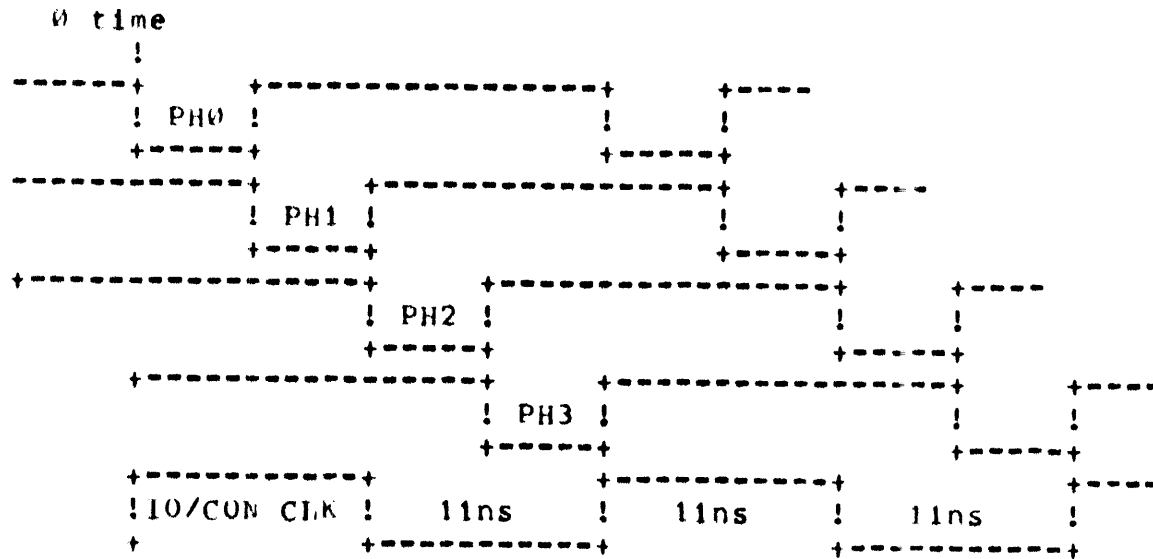
# KC10/2080 CPU PARTITIONING



10 JULY 80  
D. HOOPER  
REV 18 AUG 80

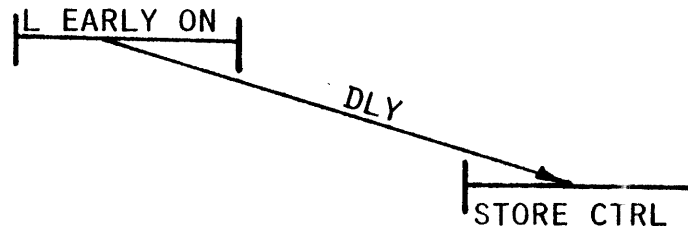
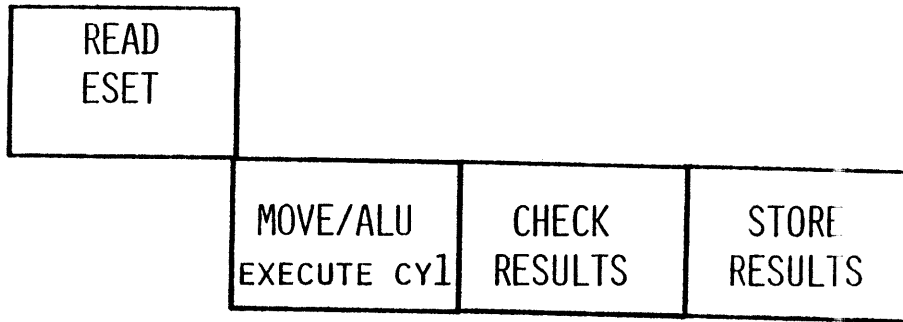
DEC COMPANY CONFIDENTIAL

# KC10/2080 CPU CLOCKS



3/6/80

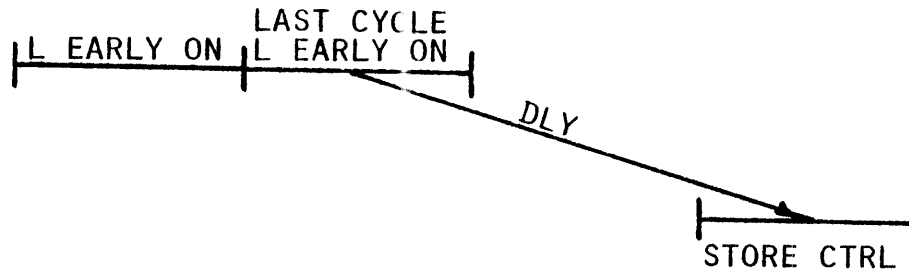
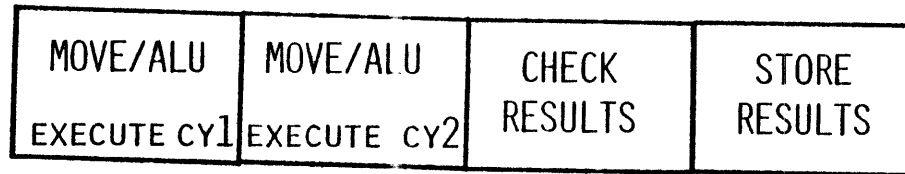
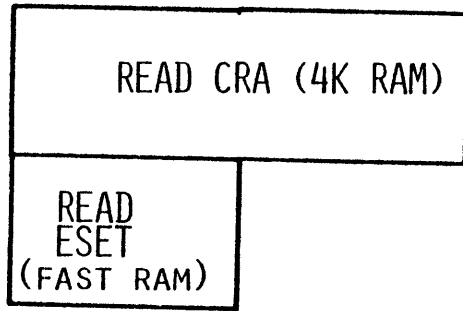
DEC COMPANY CONFIDENTIAL



22ns INSTRUCTION TIMING

3/5/80  
D.HOOPER

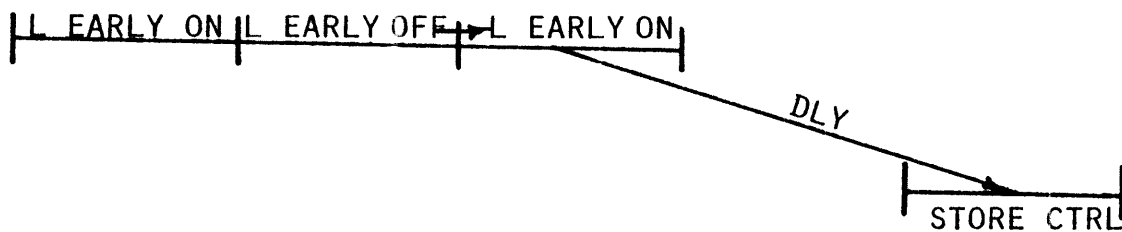
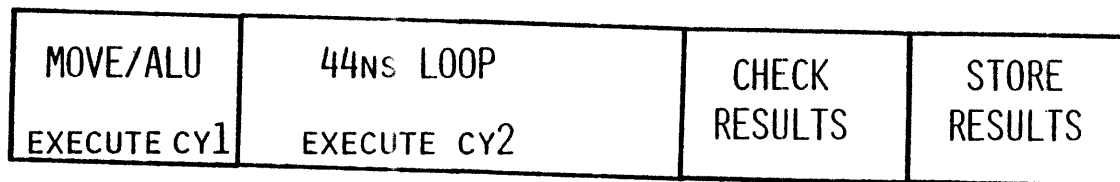
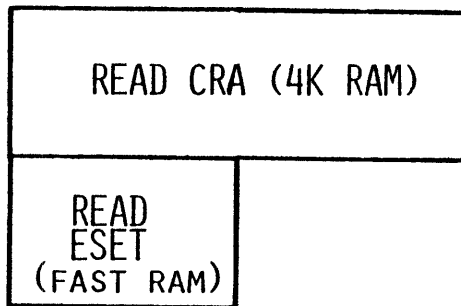
DEC COMPANY CONFIDENTIAL



44ns INSTRUCTION TIMING

3/5/80  
D.HOOPER

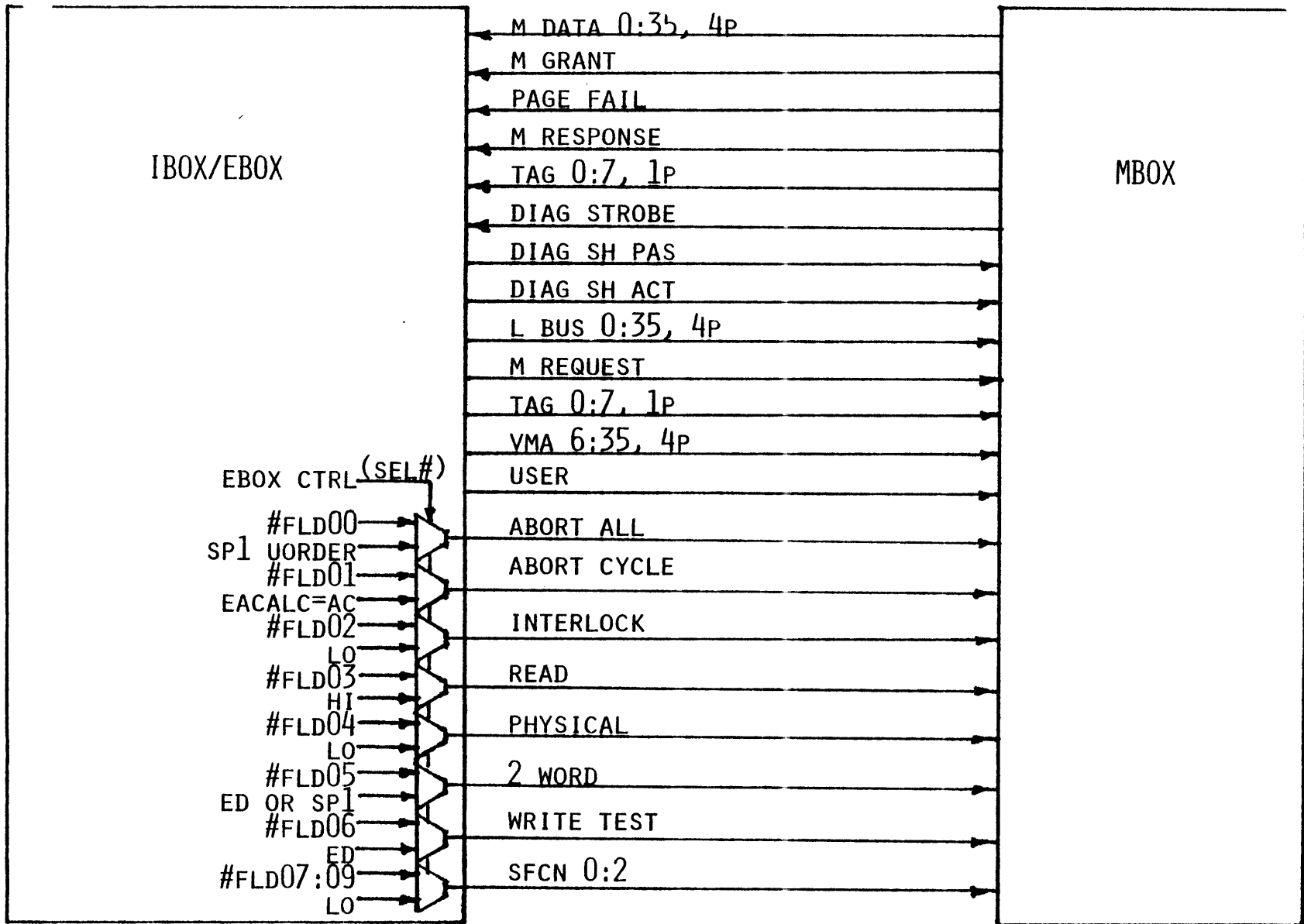
DEC COMPANY CONFIDENTIAL



66ns INSTRUCTION TIMING

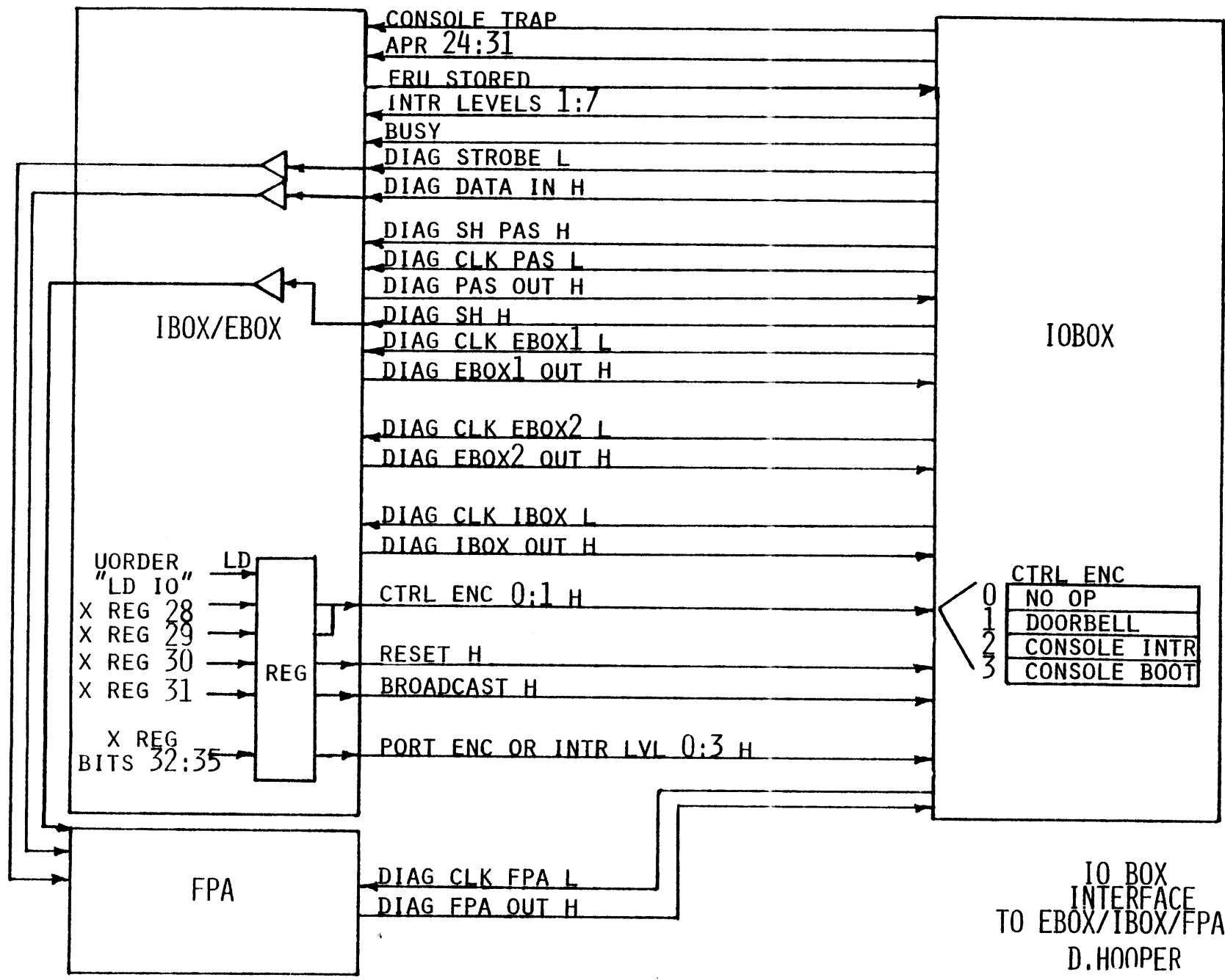
3/5/80  
D.HOOPER

DEC COMPANY CONFIDENTIAL

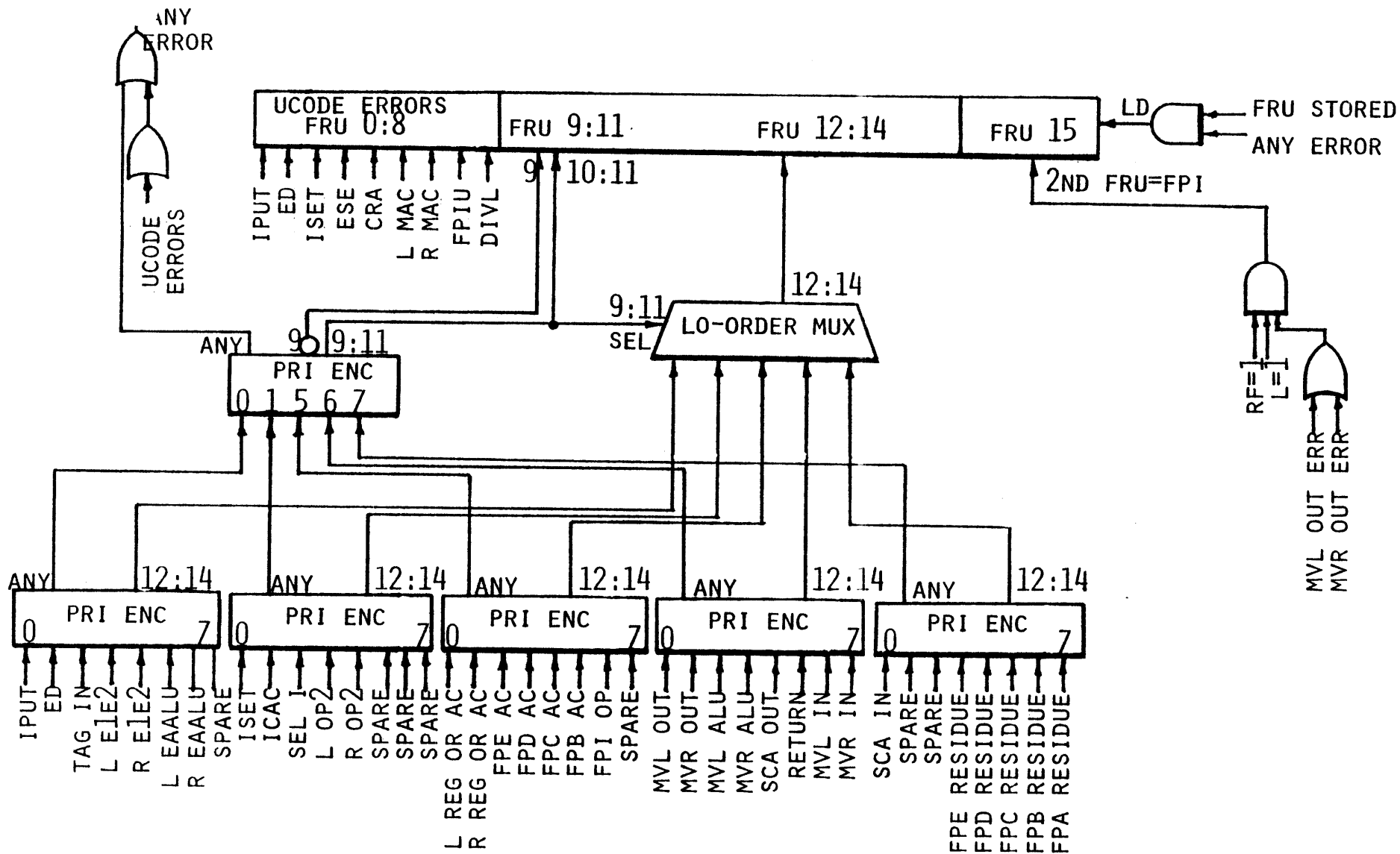


KC10/2080  
 EUNIT/MBOX INTERFACE  
 D. HOOPER  
 15AUG 80

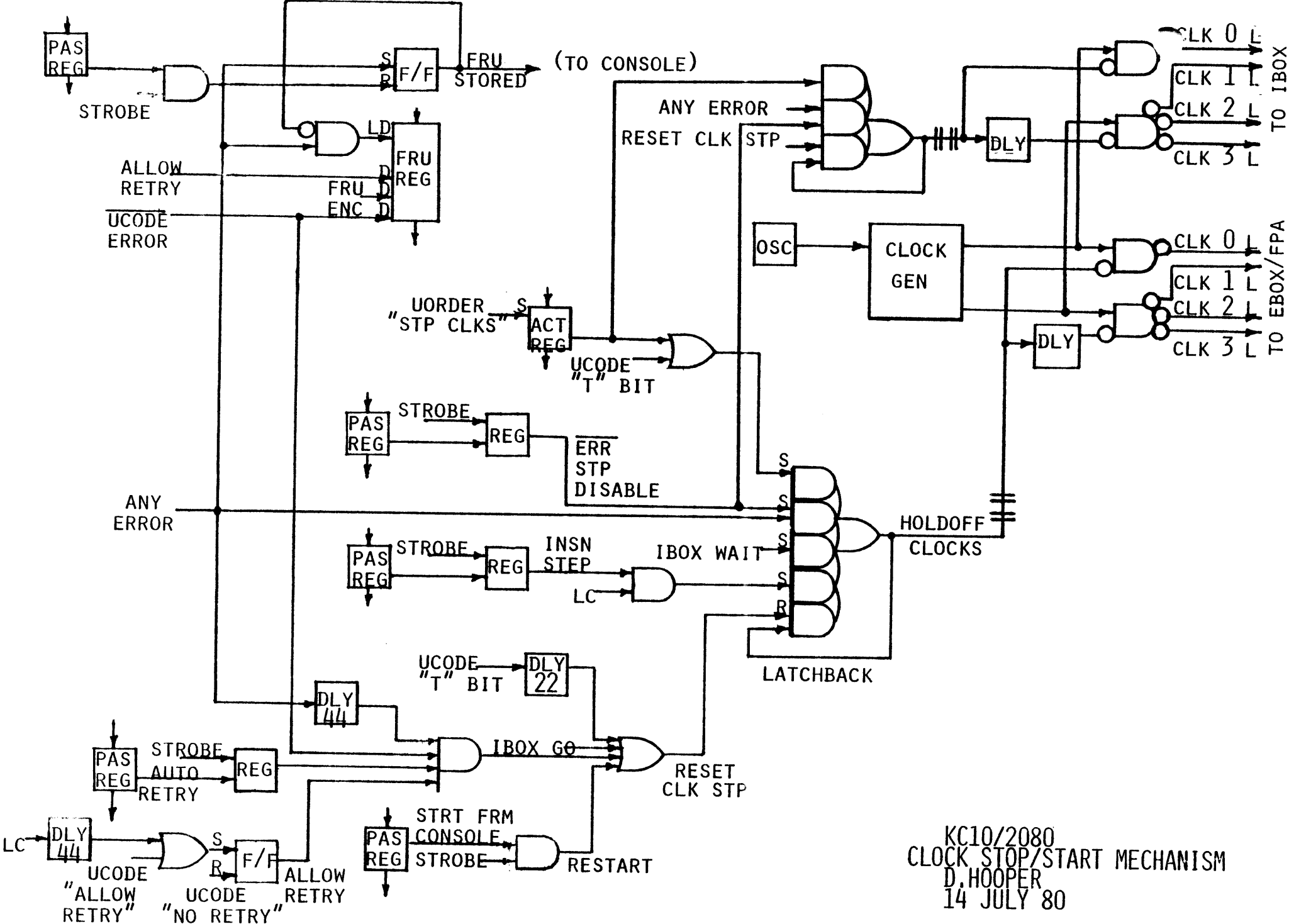
DEC COMPANY CONFIDENTIAL



IO BOX  
 INTERFACE  
 TO EBOX/IBOX/FPA  
 D. HOOPER  
 12 AUG 80



KC10/2080  
 FRU PRIORITY ENCODER  
 D. HOOPER  
 31 JULY 80

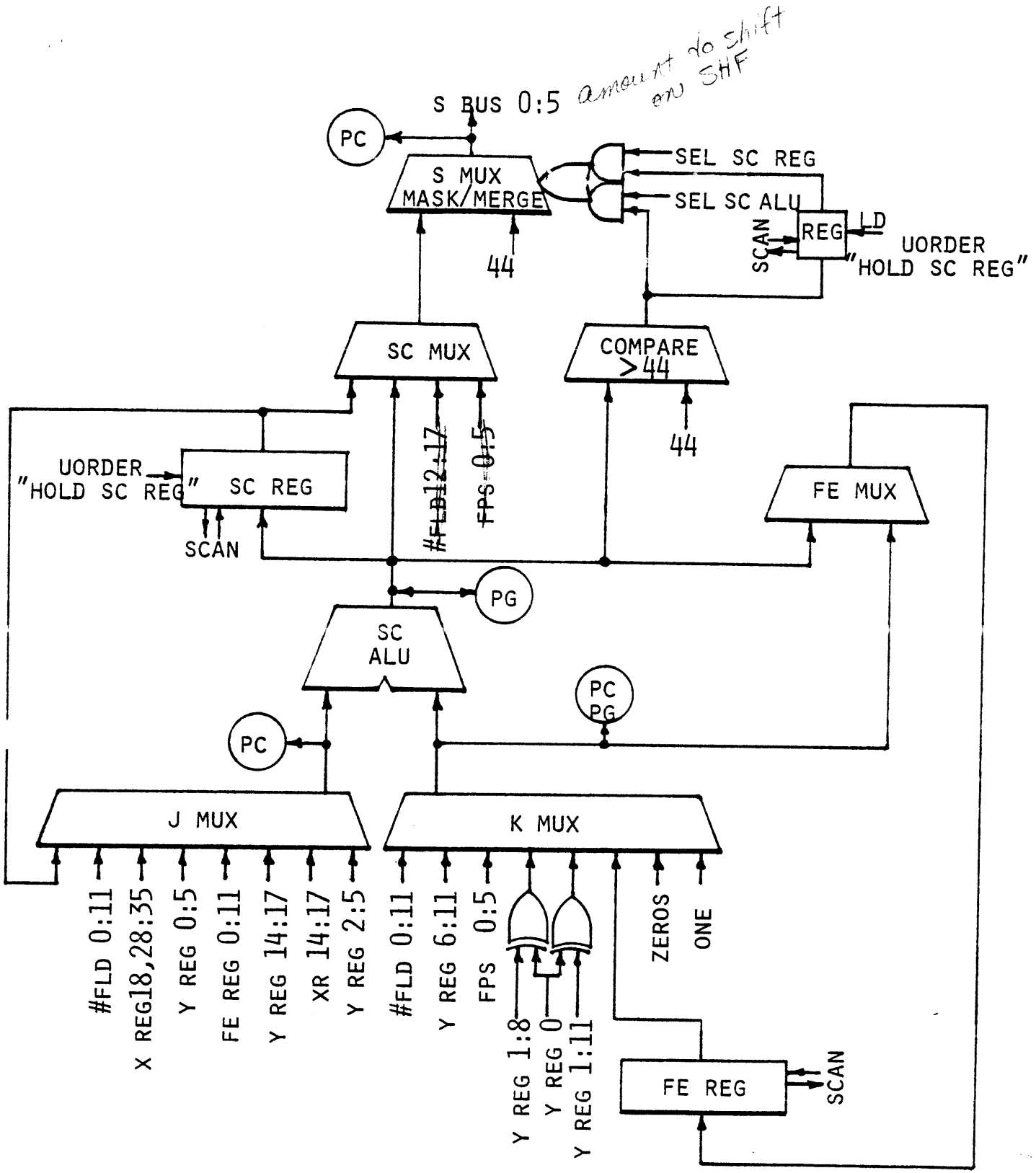


KC10/2080  
 CLOCK STOP/START MECHANISM  
 D. HOOPER  
 14 JULY 80

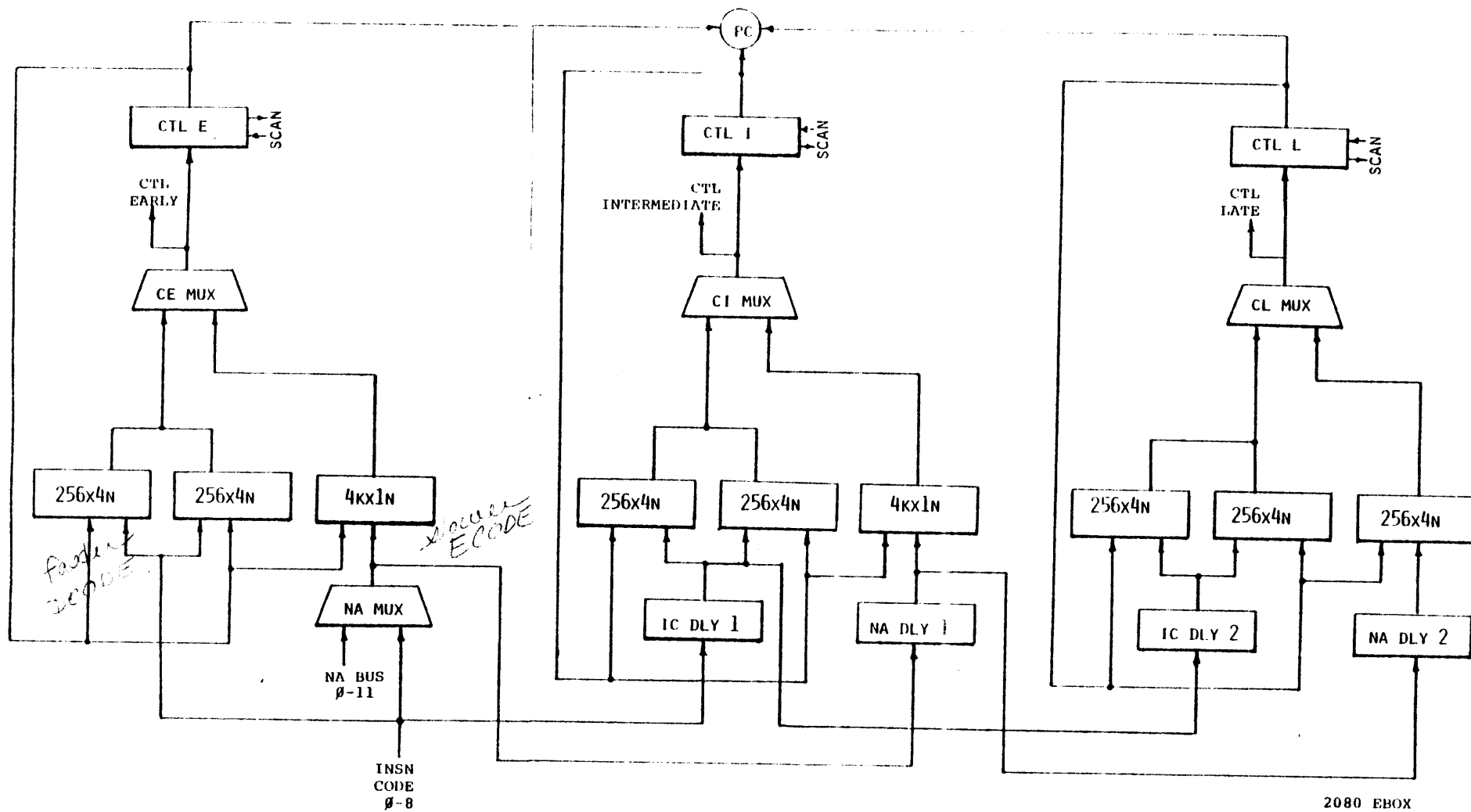
EBOX



*amount to shift on SHF*

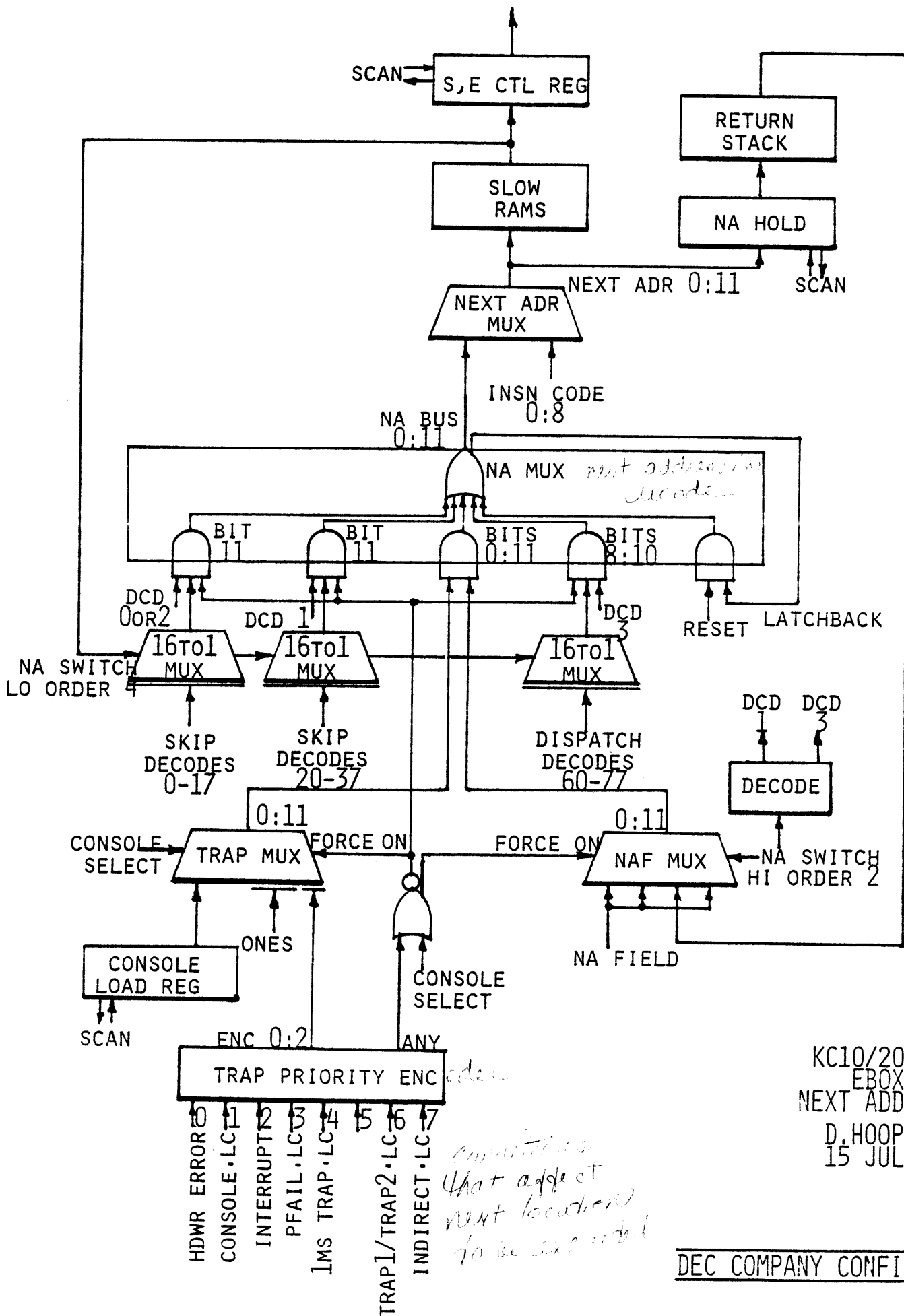


KC10/2080  
SCA  
D. HOOPER  
15 JULY 80

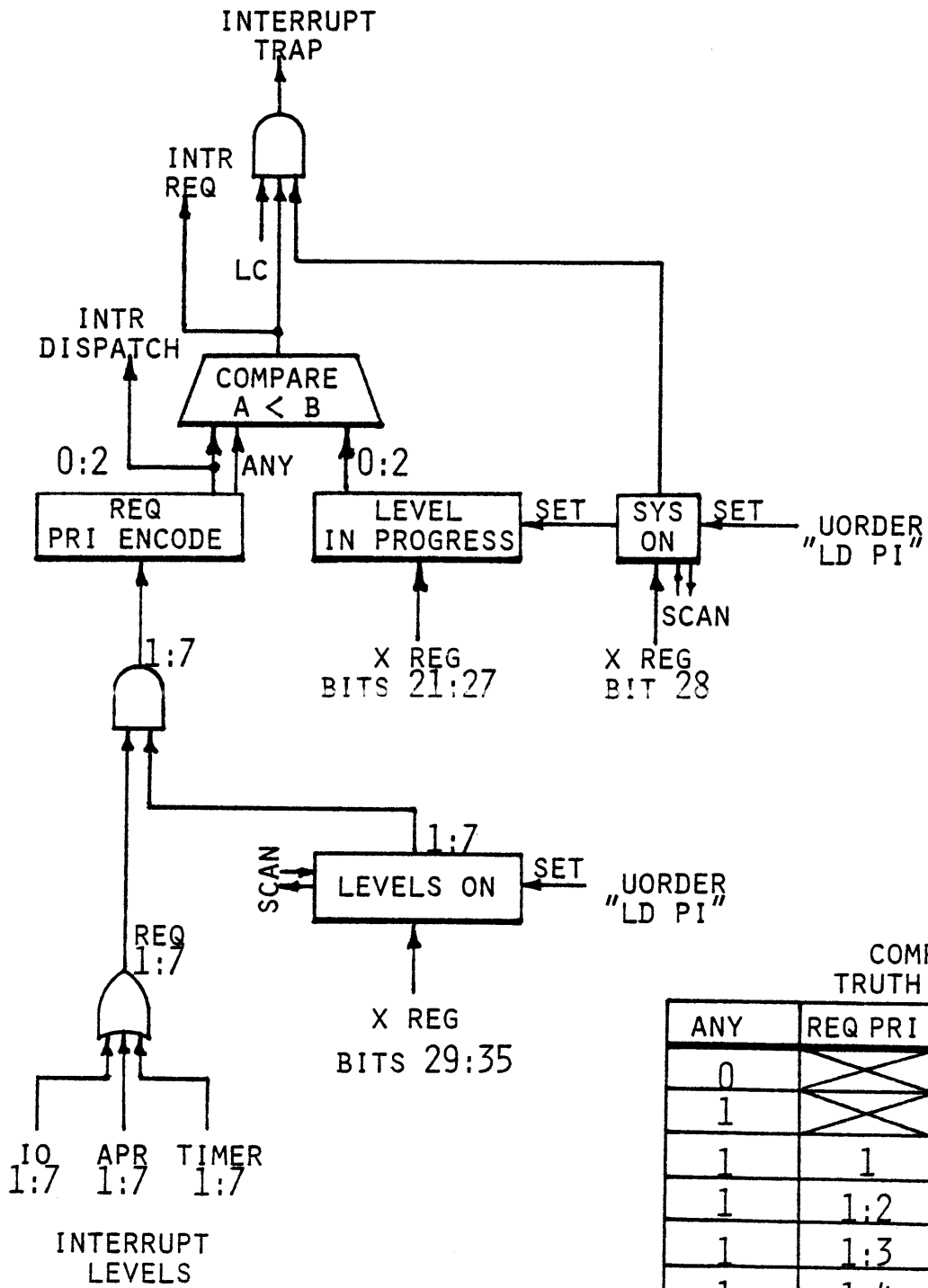


2080 EBOX  
 CTRL STORAGE  
 DON HOOPER  
 27 FEB 80

DEC COMPANY CONFIDENTIAL



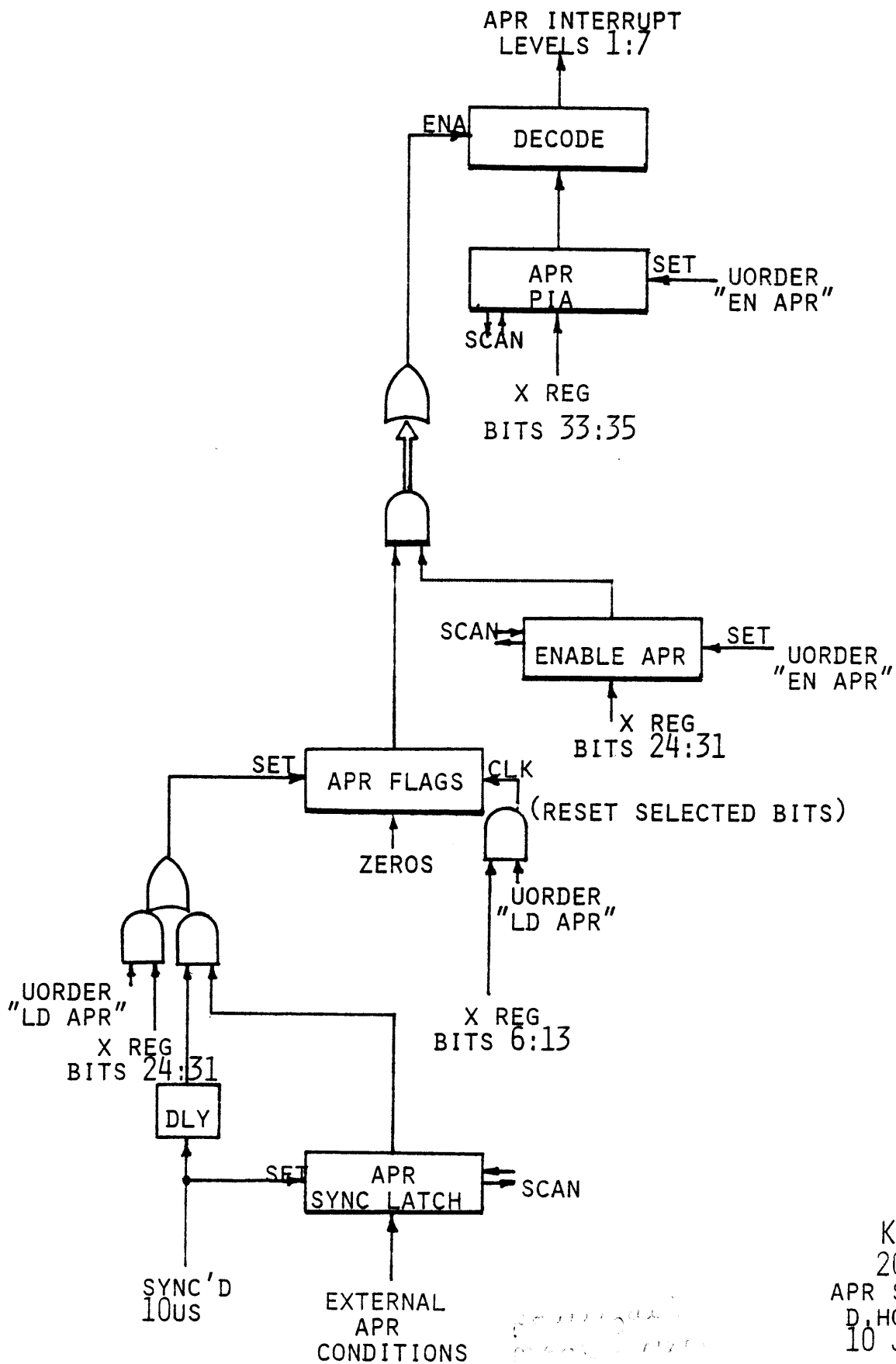
KC10/2080  
 EBOX  
 NEXT ADDRESS  
 D. HOOPER  
 15 JULY 80  
*for mode*



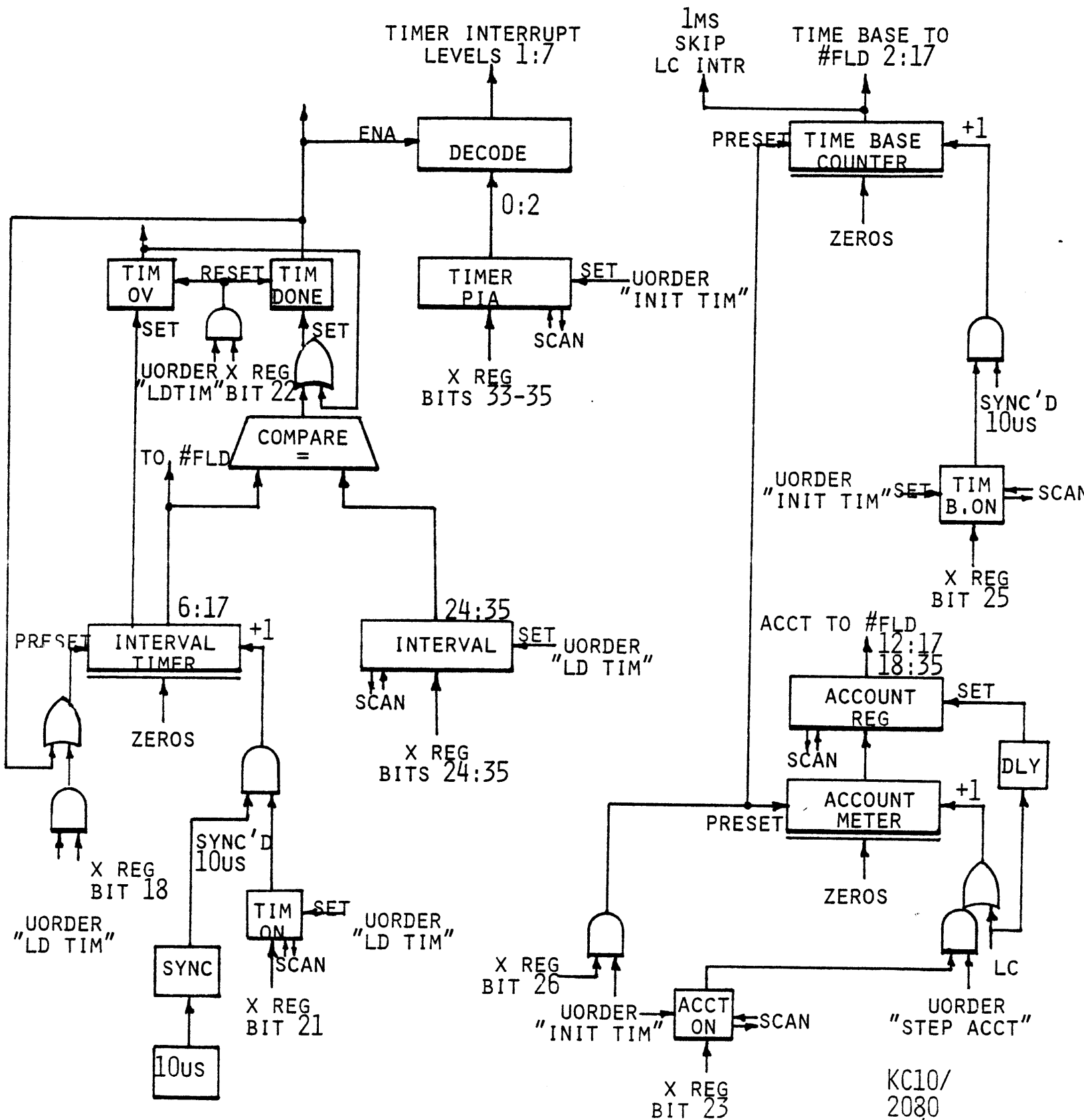
COMPARE TRUTH TABLE

| ANY | REQ PRI      | IN PROG      | A<B |
|-----|--------------|--------------|-----|
| 0   | <del>1</del> | <del>1</del> | 0   |
| 1   | <del>1</del> | 1            | 0   |
| 1   | 1            | 2            | 1   |
| 1   | 1:2          | 3            | 1   |
| 1   | 1:3          | 4            | 1   |
| 1   | 1:4          | 5            | 1   |
| 1   | 1:5          | 6            | 1   |
| 1   | 1:6          | 7            | 1   |
| 1   | 1:7          | 0            | 1   |

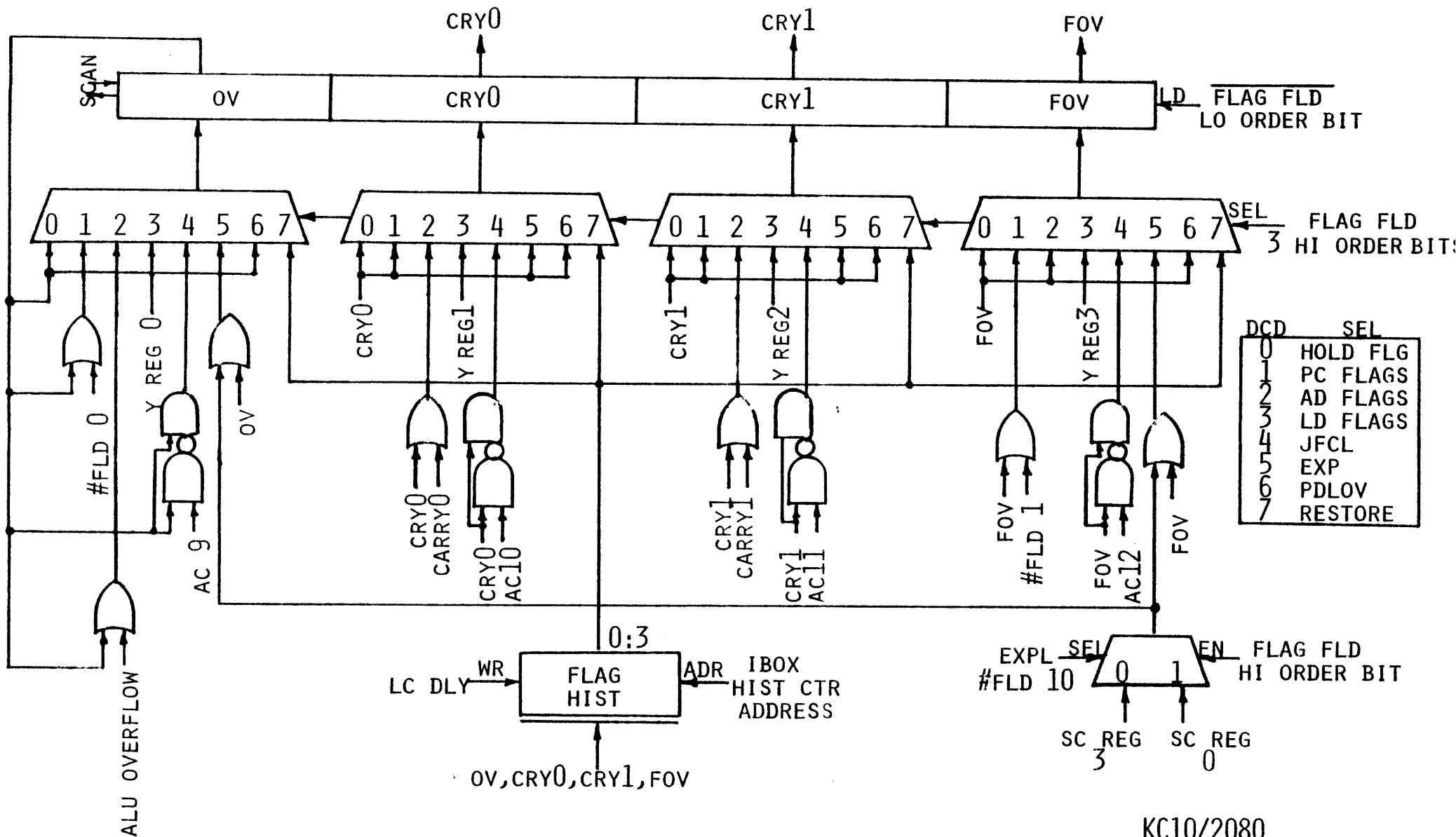
KC10/  
2080  
INTERRUPT SYSTEM  
D. HOOPER  
9 JULY 80



KC10/  
2080  
APR SYSTEM  
D. HOOPER  
10 JULY 80

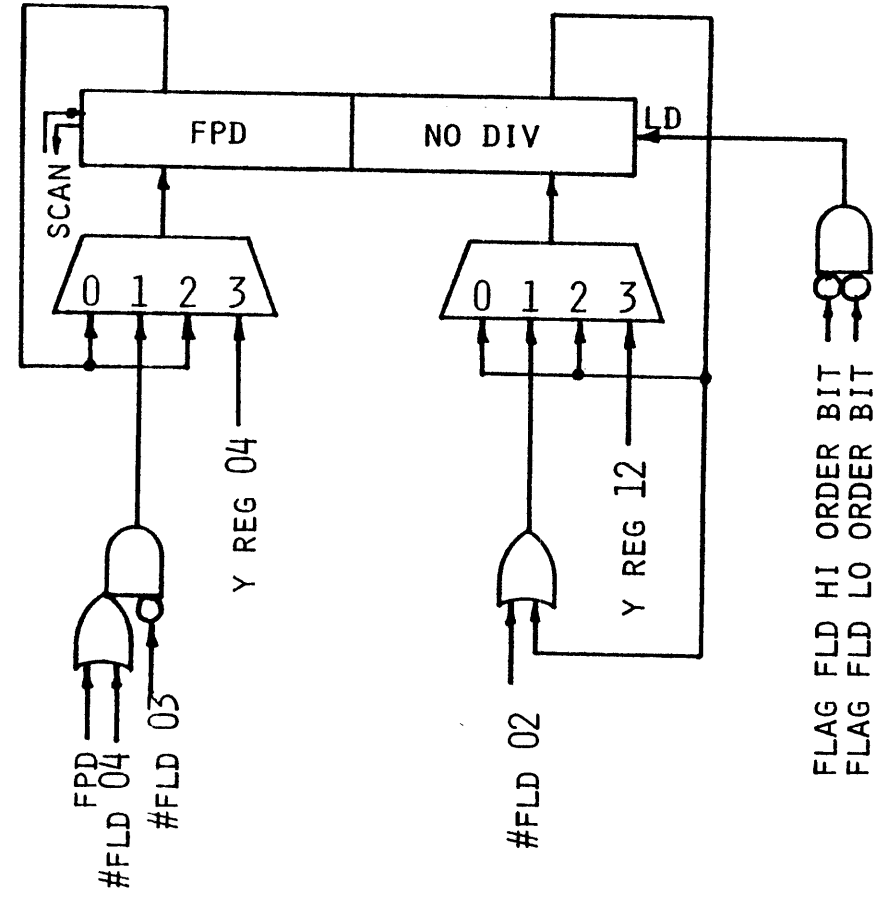
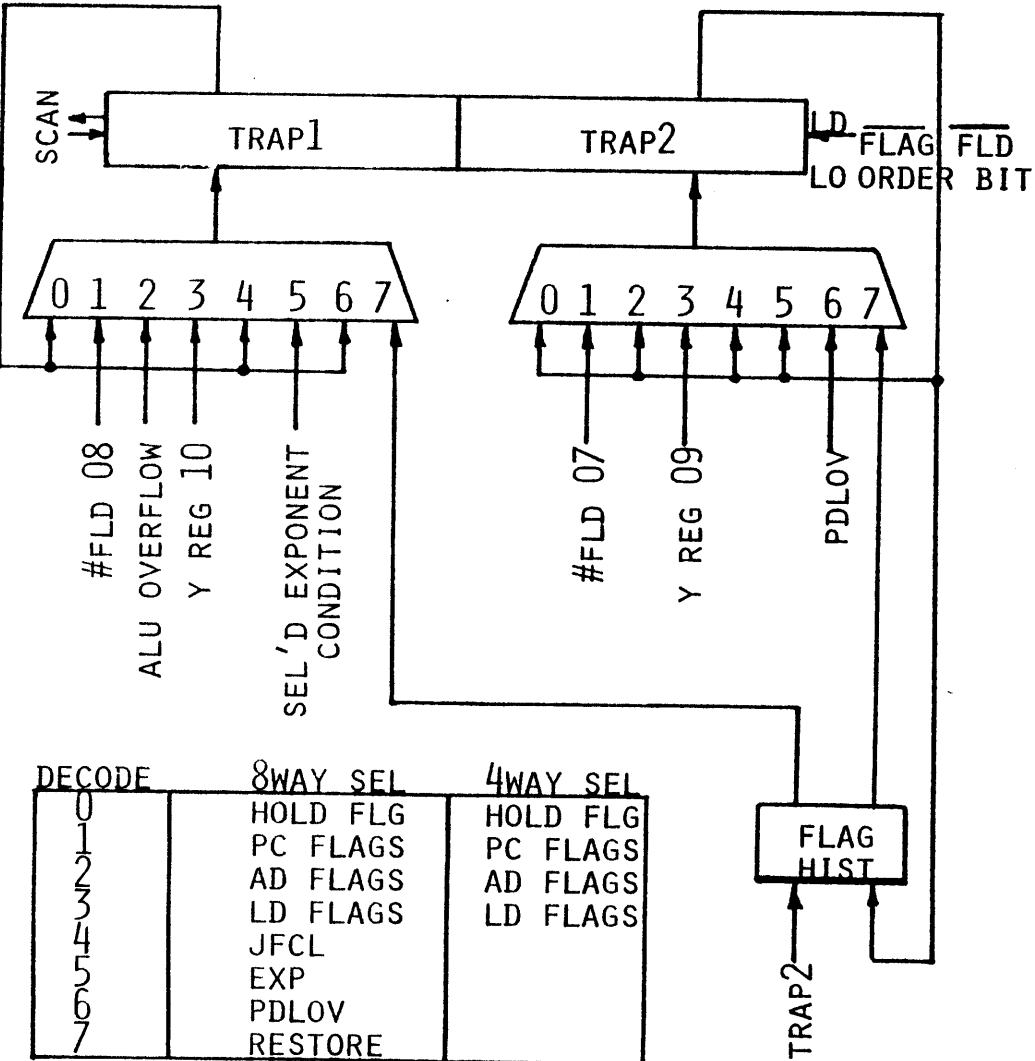


KC10/  
 2080  
 TIMERS/METERS  
 D. HOOPER  
 10 JULY 80

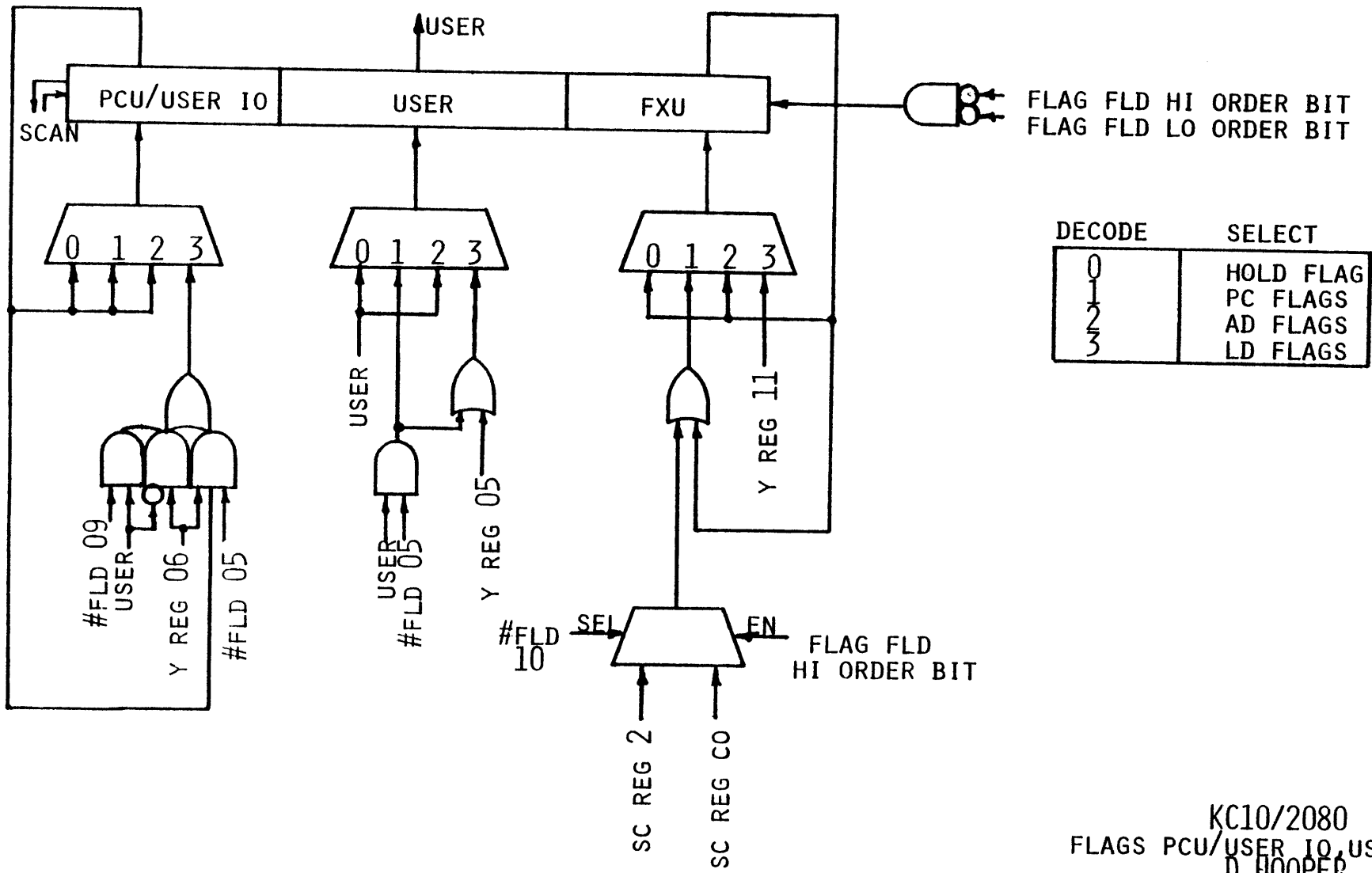


DEC COMPANY CONFIDENTIAL

KC10/2080  
 FLAGS OV, CRY0, CRY1, FOV  
 D. HOOPER  
 16 JULY 80



KC10/2080  
 FLAGS TRAP1, TRAP2, FPD, NO DIV  
 D. HOOPER  
 22 JULY 80

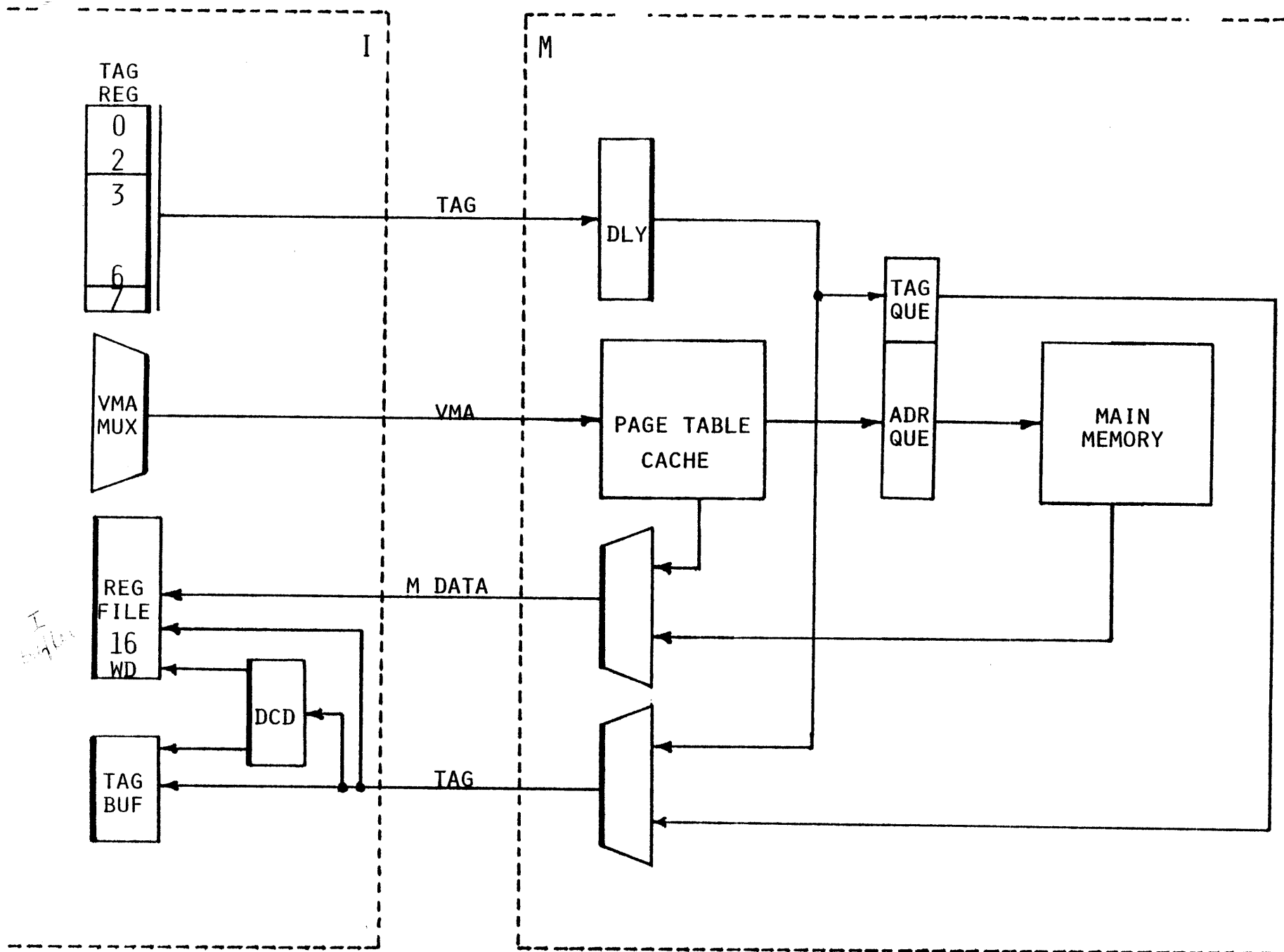


| DECODE | SELECT    |
|--------|-----------|
| 0      | HOLD FLAG |
| 1      | PC FLAGS  |
| 2      | AD FLAGS  |
| 3      | LD FLAGS  |

KC10/2080  
 FLAGS PCU/USER IO, USER, FXU  
 D. HOOPER  
 22 JULY 80

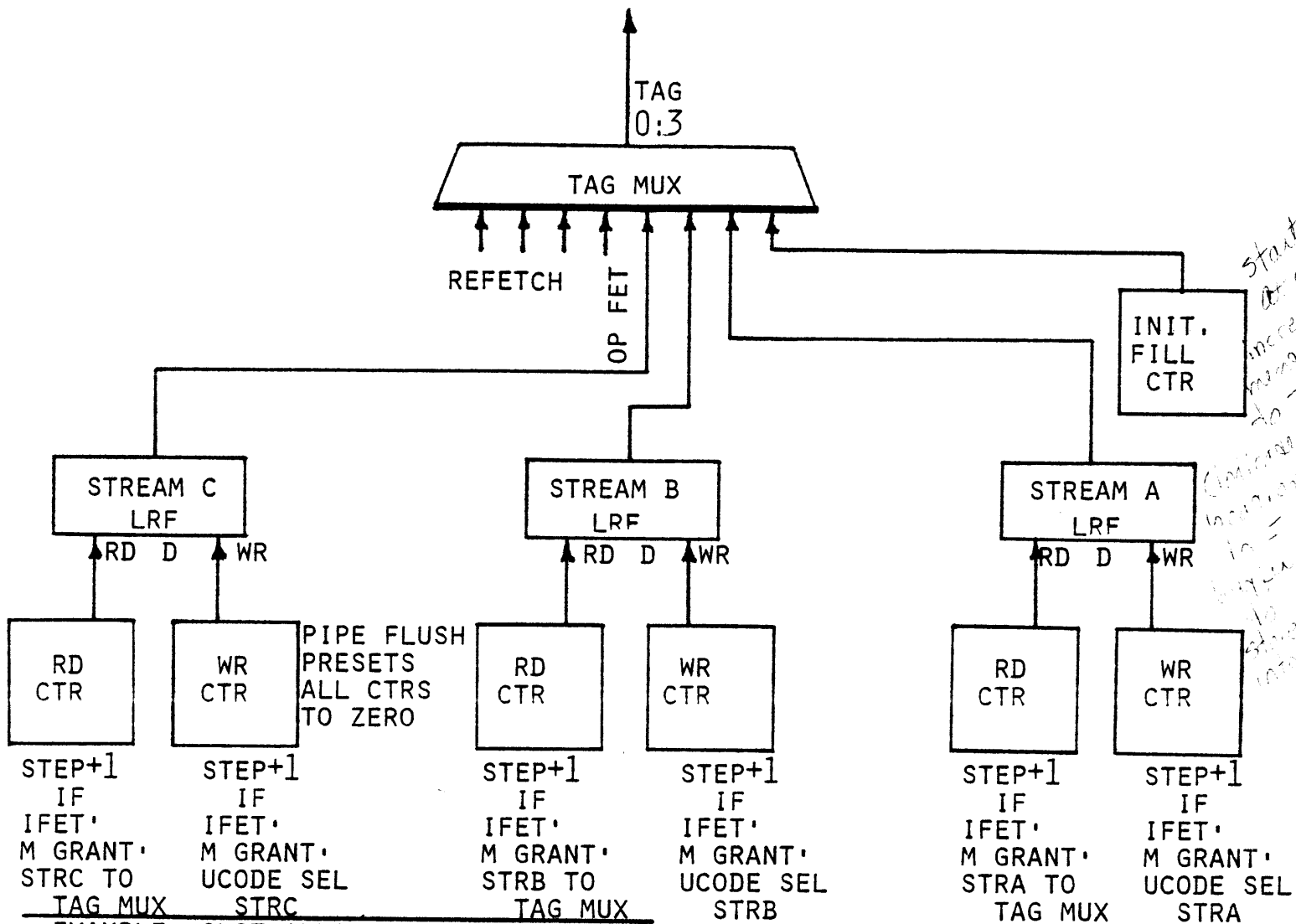
DEC COMPANY CONFIDENTIAL

IBOX

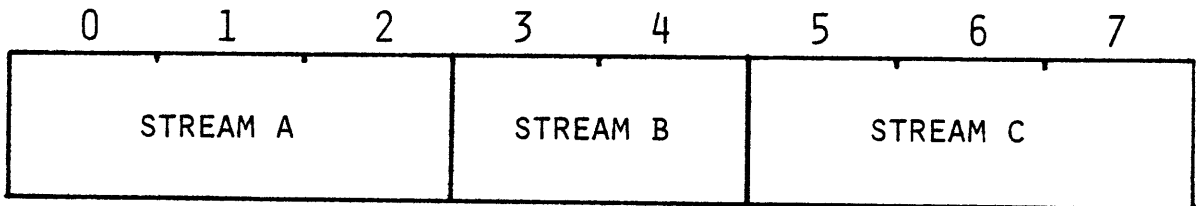


2080 TAG SYSTEM

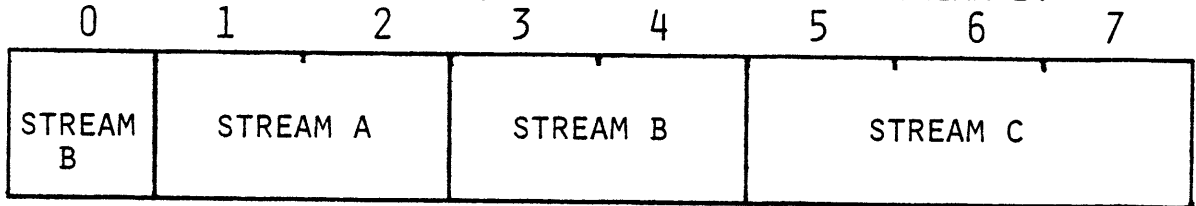
D.HOOPER 3/5/80



EXAMPLE: SLOT ALLOCATION BY STREAM



STREAM A IS CANCELLED BY EBOX, IBOX FETCHES INTO STREAM B:



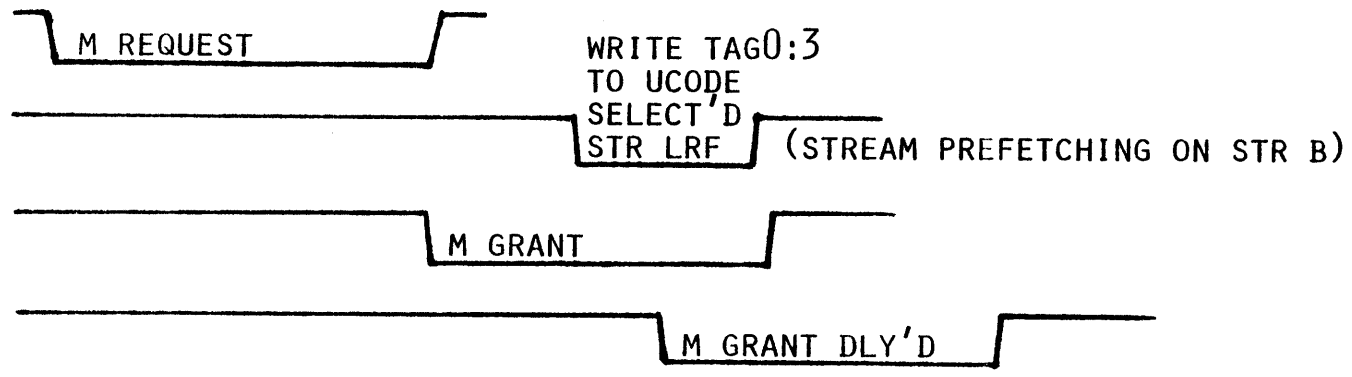
KC10/2080 LRF SYSTEM

D. HOOPER  
3/5/80 REV15AUG80



INPUT EARLY CTRL

VMA+TAG+CTRL TO M



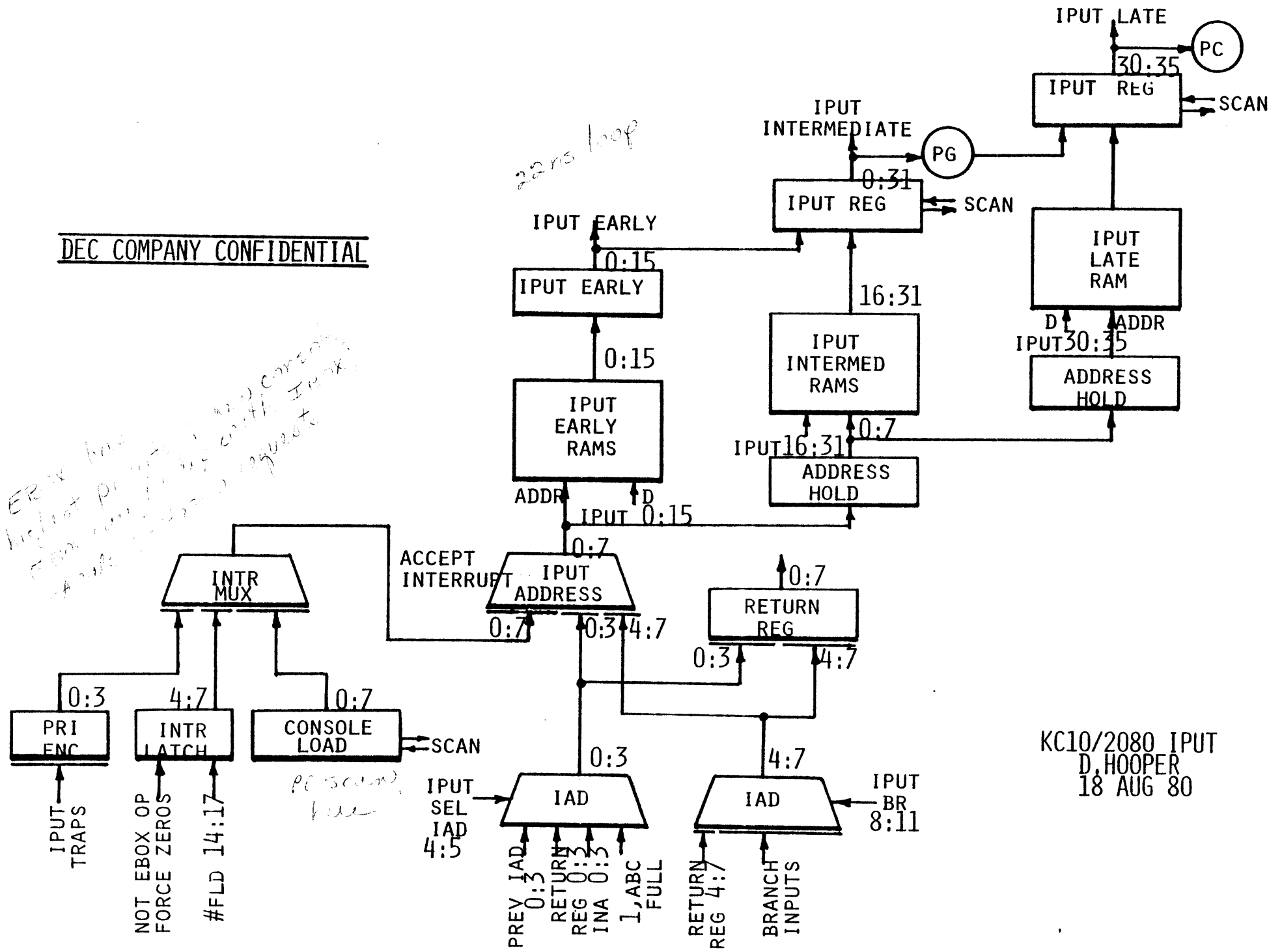
WR CTR +1 STR B

RD CTR +1 STR A (PRIOR TO THIS, EBOX HAS CANCELLED STR A, CAUSING SELECTION OF STR A TO NEXT ITAG)

DEC COMPANY CONFIDENTIAL

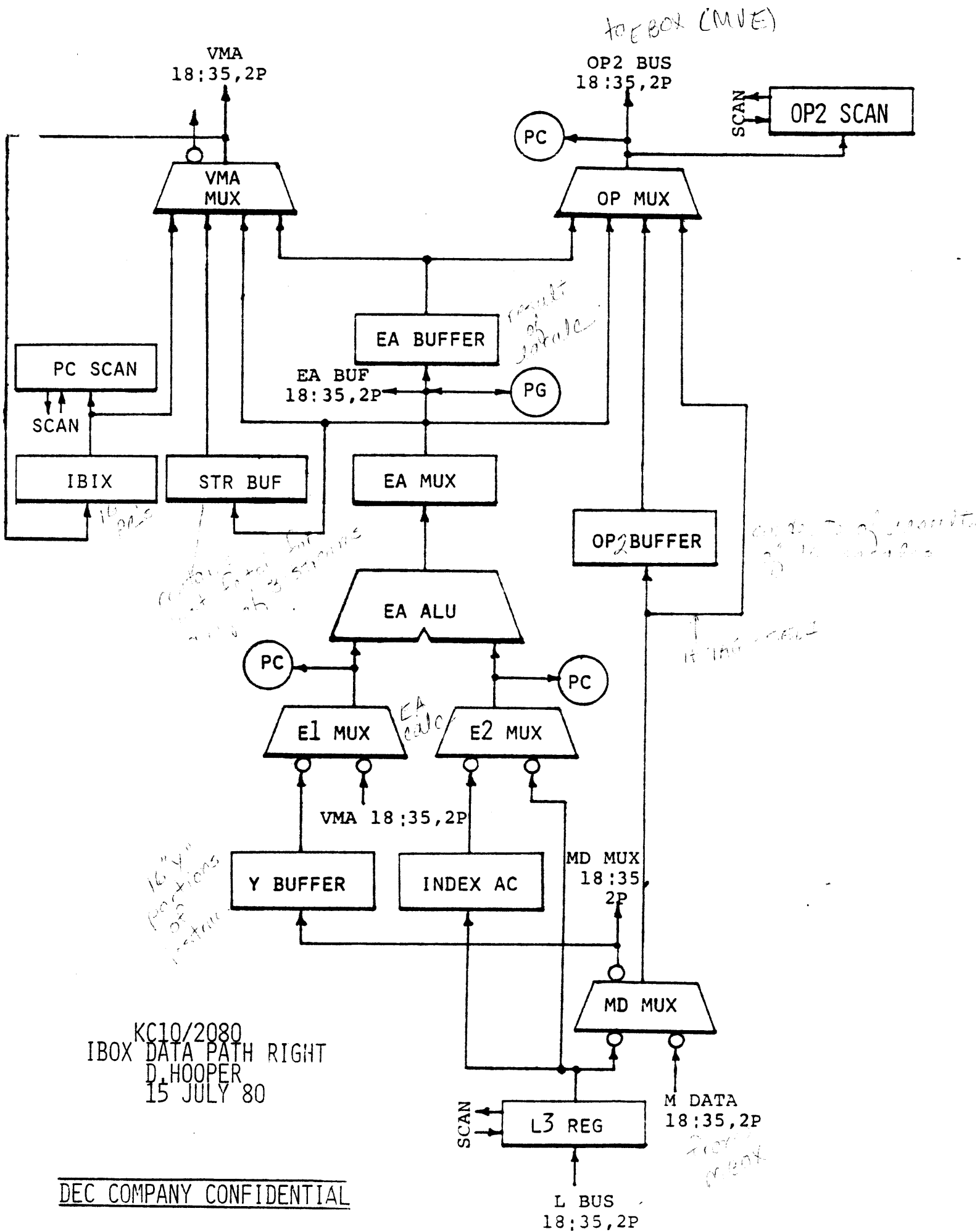
KC10/2080  
 LRF READ-WRITE TIMING  
 D. HOOPER  
 15 AUG 80

DEC COMPANY CONFIDENTIAL

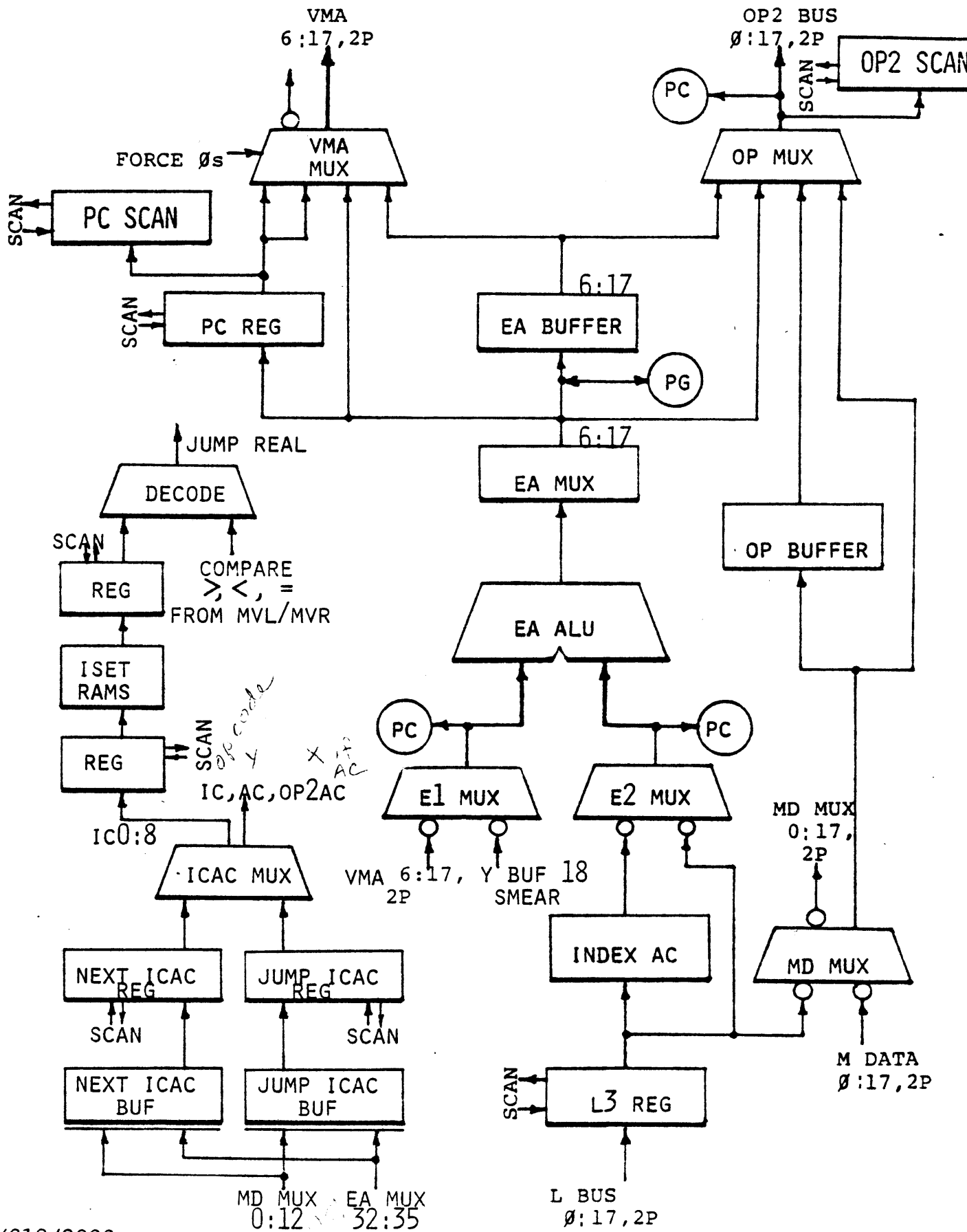


*ERX line  
highest priority with IENK  
from console request*

KC10/2080 IPUT  
D. HOOPER  
18 AUG 80

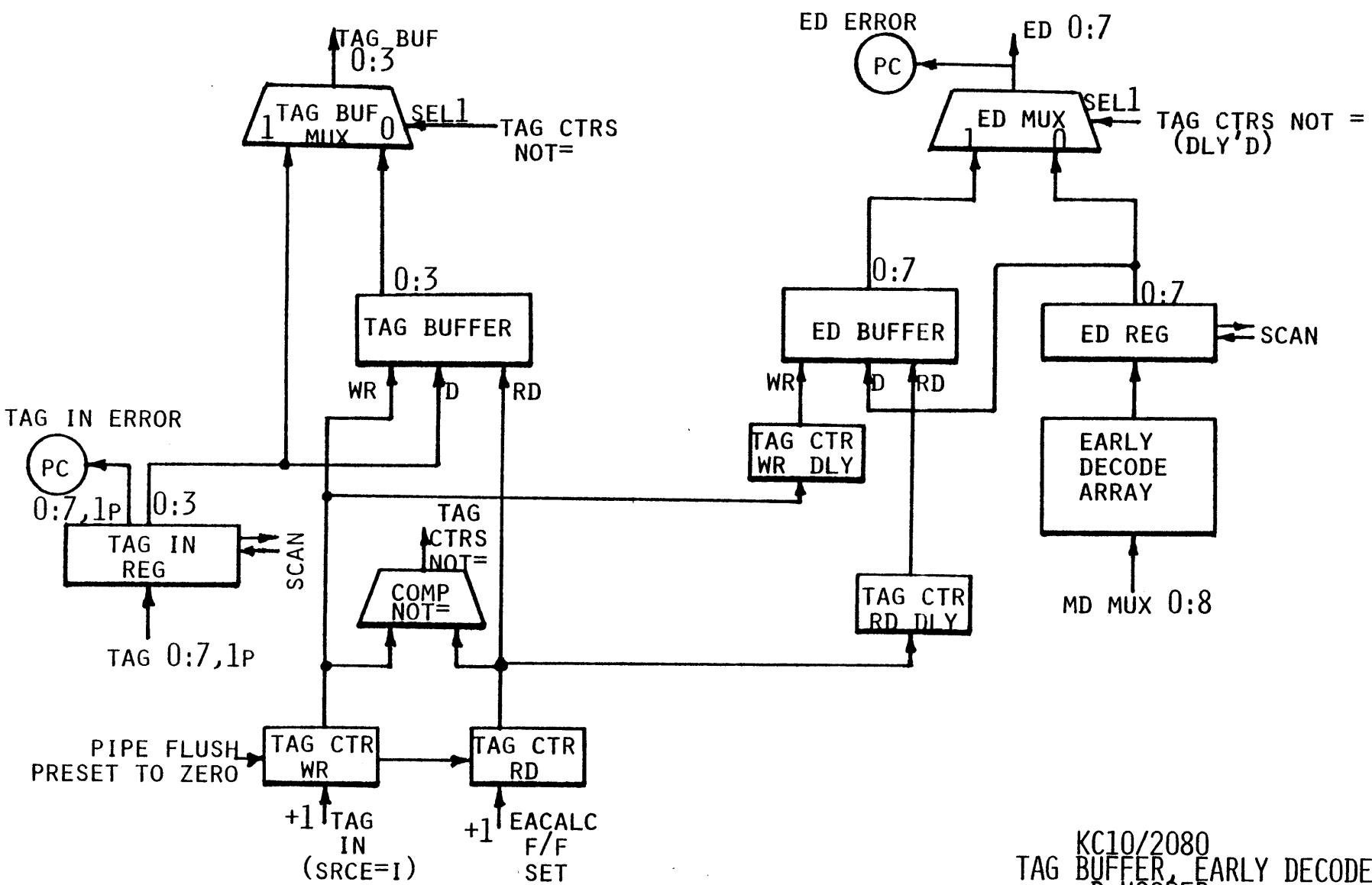


KC10/2080  
 IBOX DATA PATH RIGHT  
 D. HOOPER  
 15 JULY 80



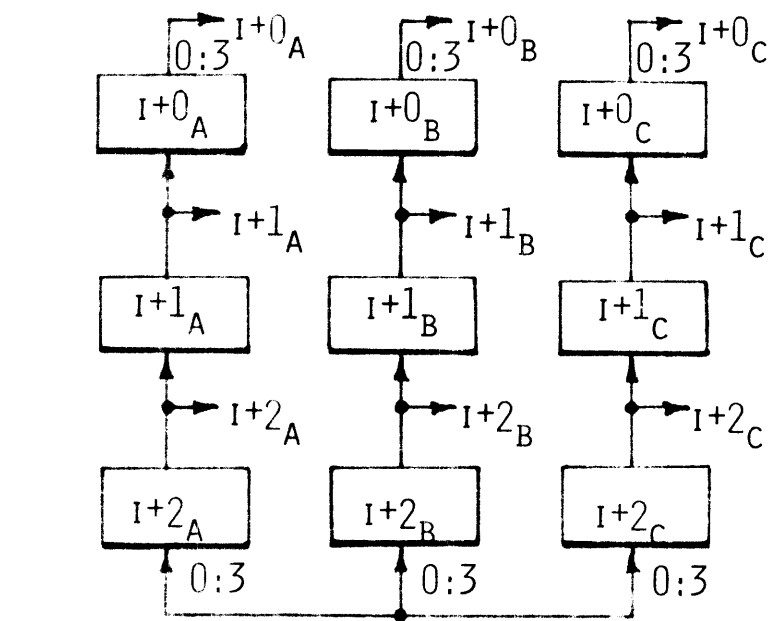
IBOX KC10/2080  
 DATA PATH LEFT  
 D. HOOPER  
 15 JULY 80

DEC COMPANY CONFIDENTIAL



KC10/2080  
TAG BUFFER, EARLY DECODE  
D. HOOPER  
18 AUG 80

next instruction on  
same box

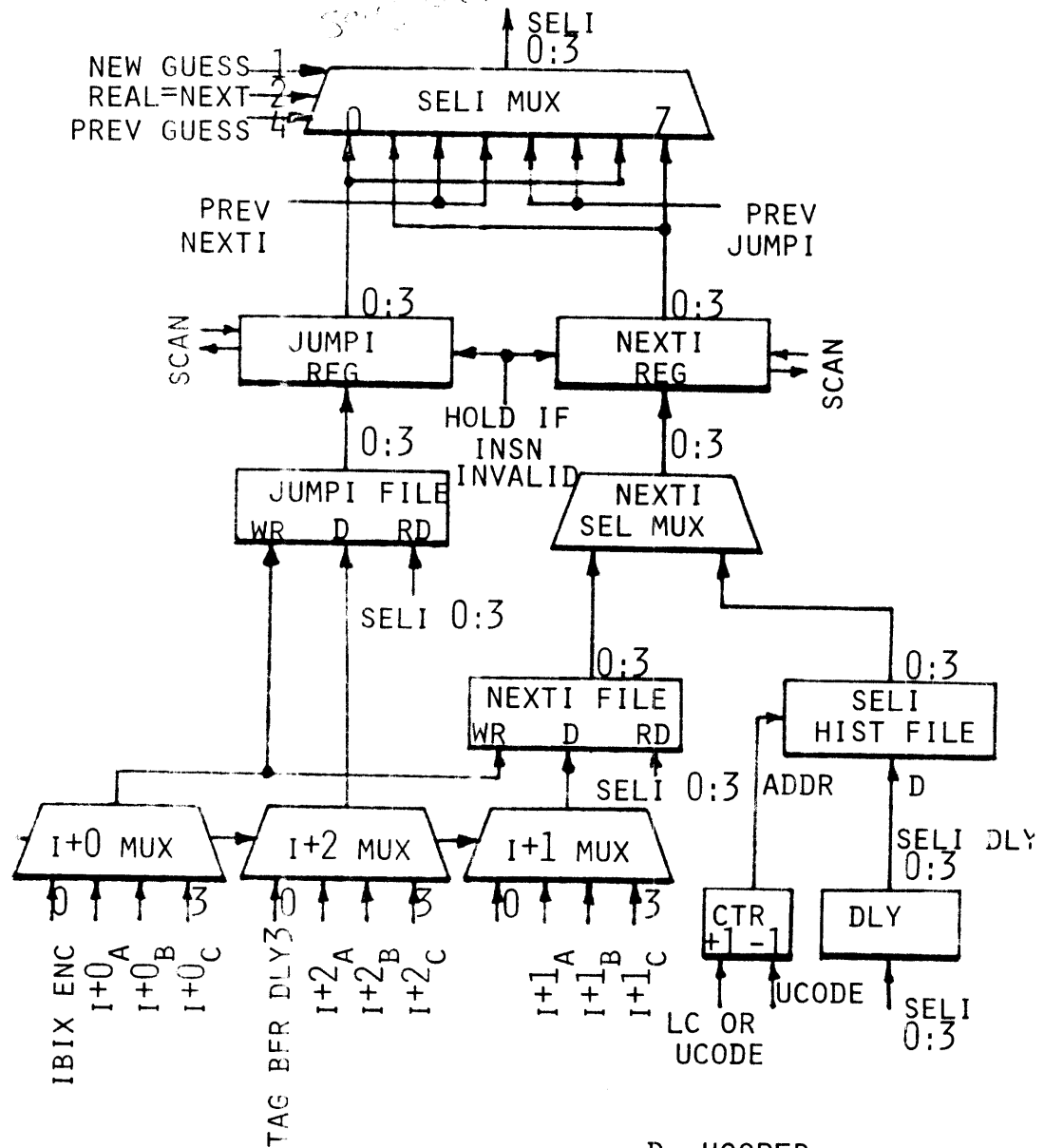


PREV GUESS NXT  
REAL NEXT  
NEW GUESS NXT

|     |              |
|-----|--------------|
| 000 | SEL JMP      |
| 001 | SEL NXT      |
| 010 | SEL PREV NXT |
| 011 | SEL PREV NXT |
| 100 | SEL PREV JMP |
| 101 | SEL PREV JMP |
| 110 | SEL JMP      |
| 111 | SEL NXT      |

TAG OUT 0:3

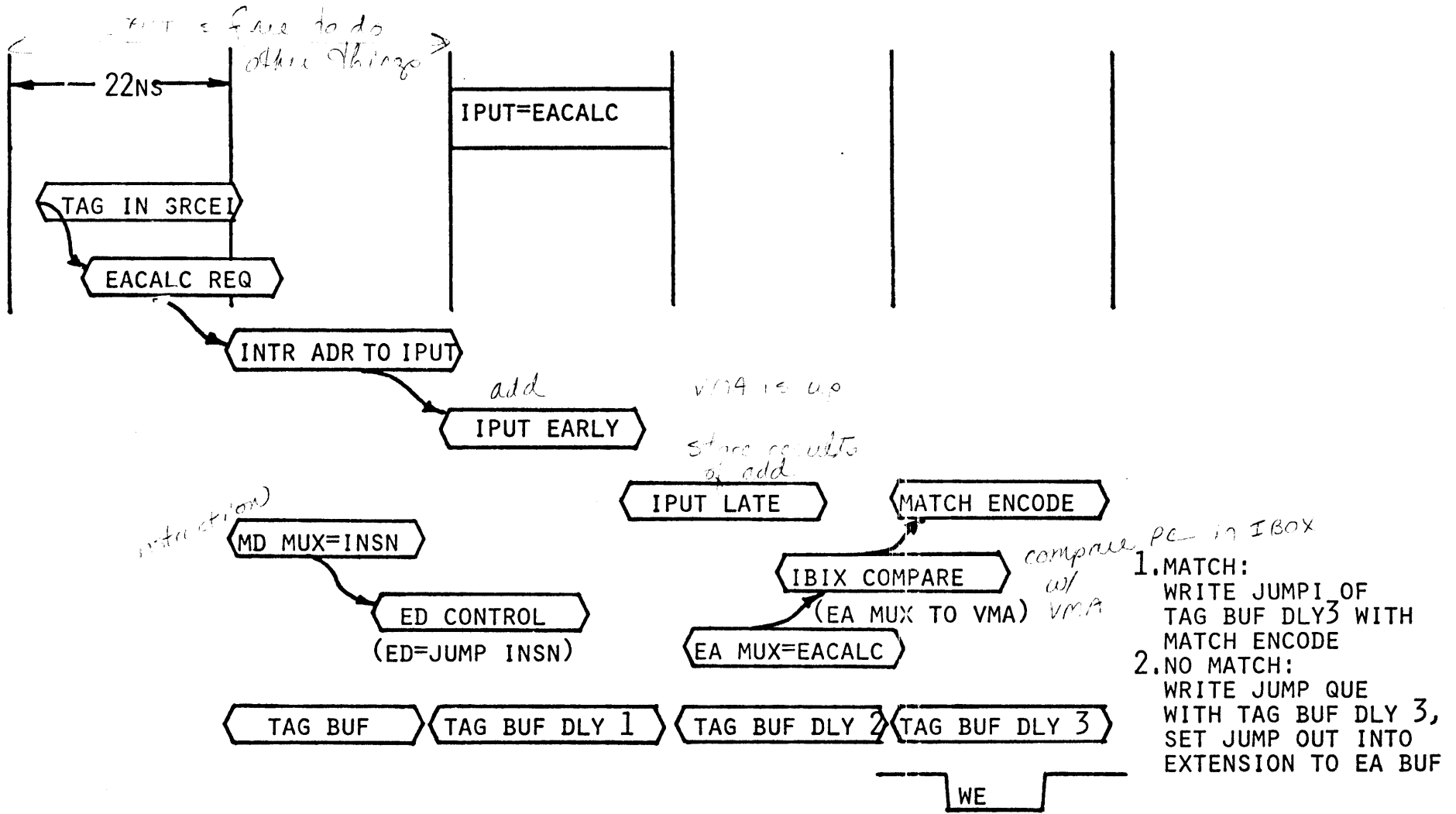
STR SEL



*Shows how it fixes bad guesses - keeps history of previous guesses*

KC10/2080 SETUP OF NEXT OR SKIP/JUMP INSTRUCTION

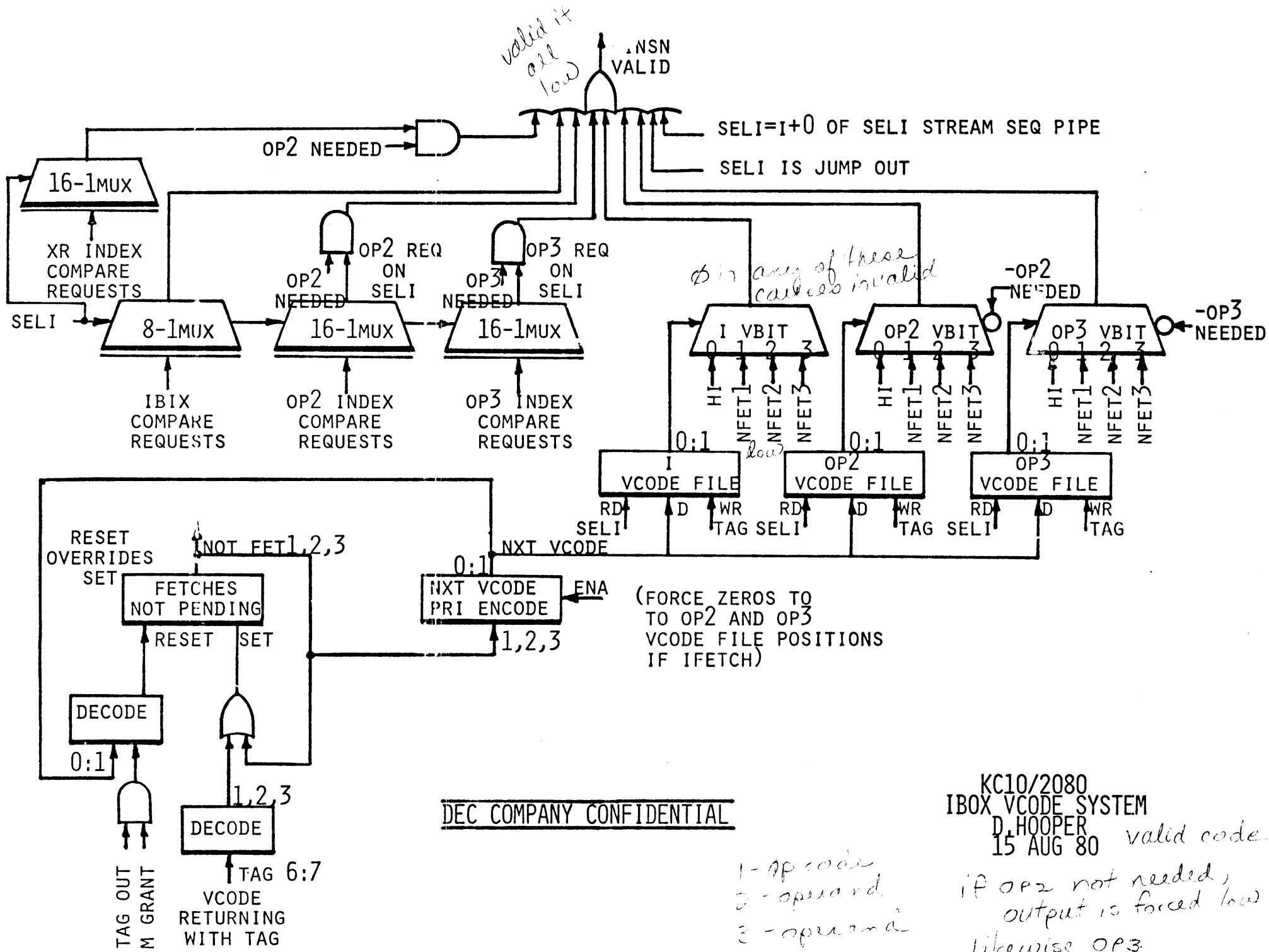
D. HOOPER  
15 AUG 80



1. MATCH:  
WRITE JUMPI\_OF TAG BUF DLY3 WITH MATCH ENCODE
2. NO MATCH:  
WRITE JUMP QUE WITH TAG BUF DLY 3, SET JUMP OUT INTO EXTENSION TO EA BUF

DEC COMPANY CONFIDENTIAL

KC10/2080  
EACALC OF JUMP  
D. HOOPER  
19 AUG 80



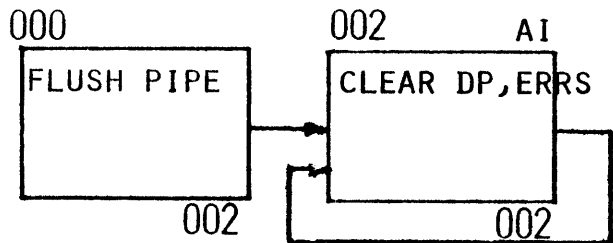
DEC COMPANY CONFIDENTIAL

KC10/2080  
IBOX VCODE SYSTEM  
D. HOOPER  
15 AUG 80

valid code

1 - opcode  
2 - operand  
3 - operand

if OP2 not needed, output is forced low likewise OP3

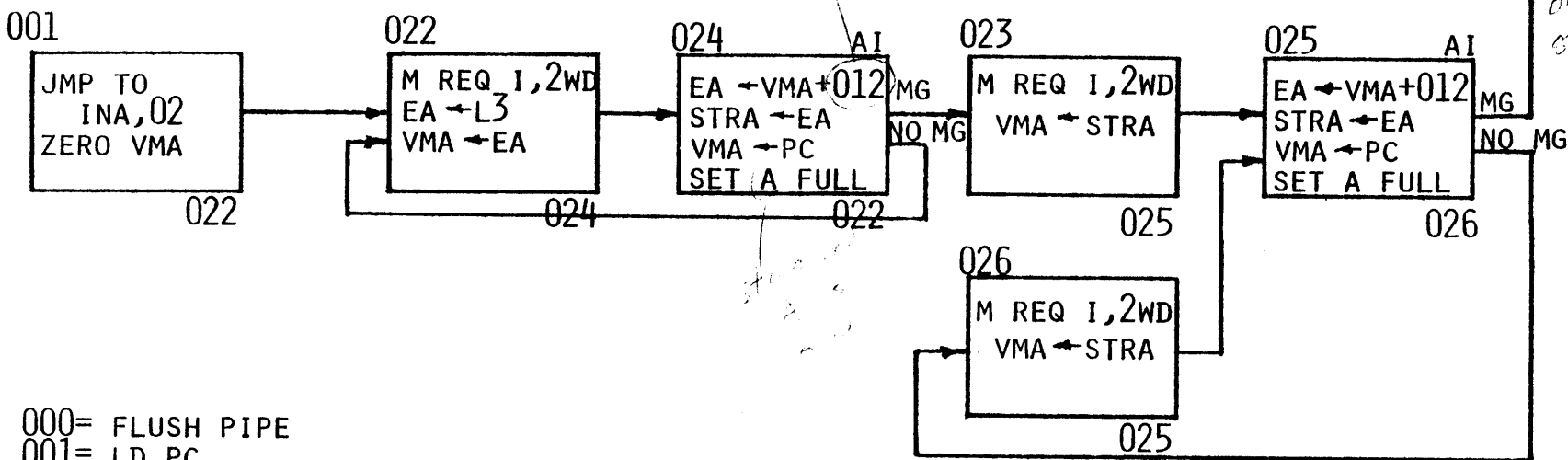


*difficult to read (no program)  
 1100 0100 0100 0100  
 1100 0100 0100 0100  
 1100 0100 0100 0100*

| ABC | IAD |
|-----|-----|
| 000 | 200 |
| 001 | 220 |
| 010 | 240 |
| 011 | 260 |
| 100 | 300 |
| 101 | 320 |
| 110 | 340 |
| 111 | 360 |

*insert address*

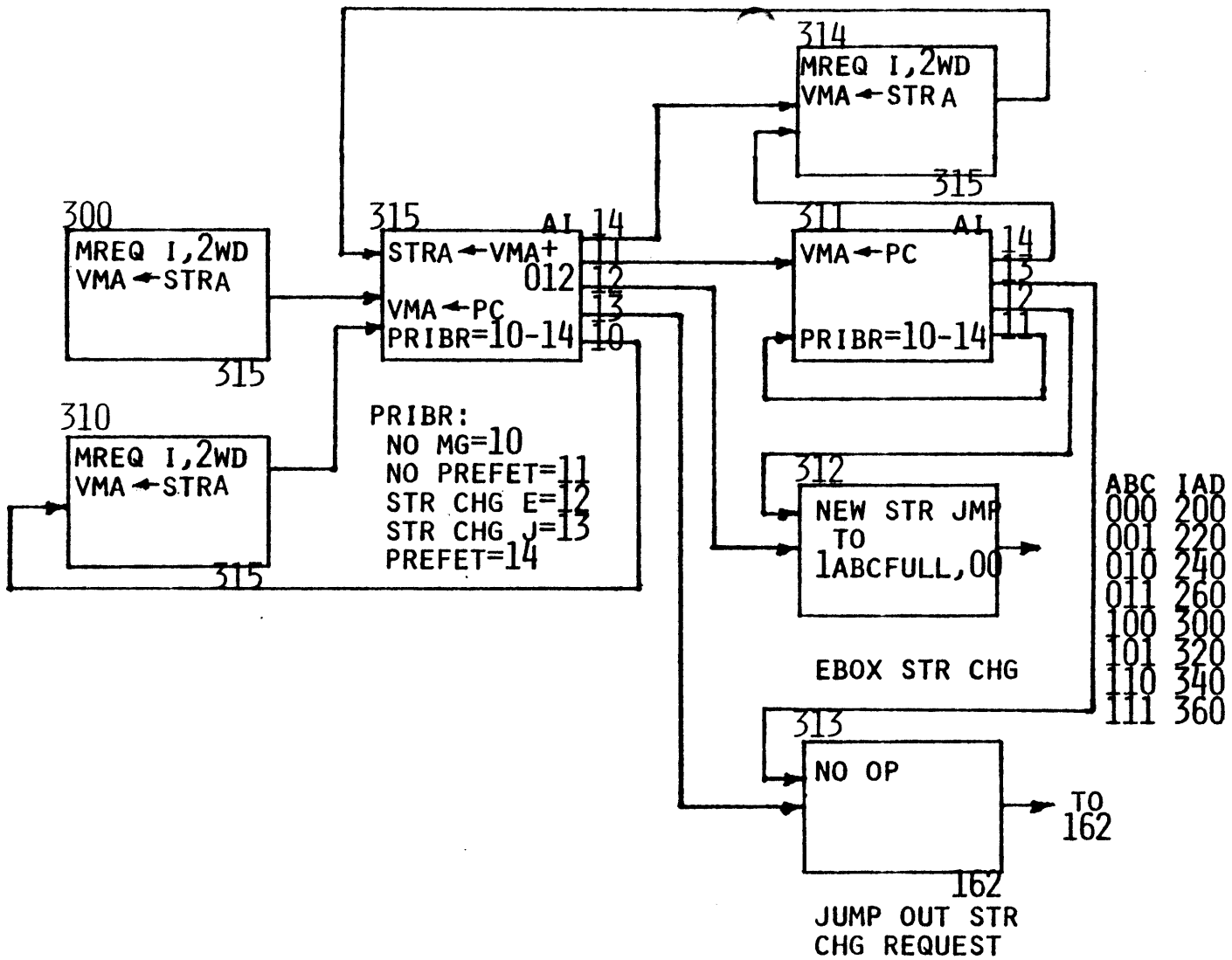
*if stream is full*



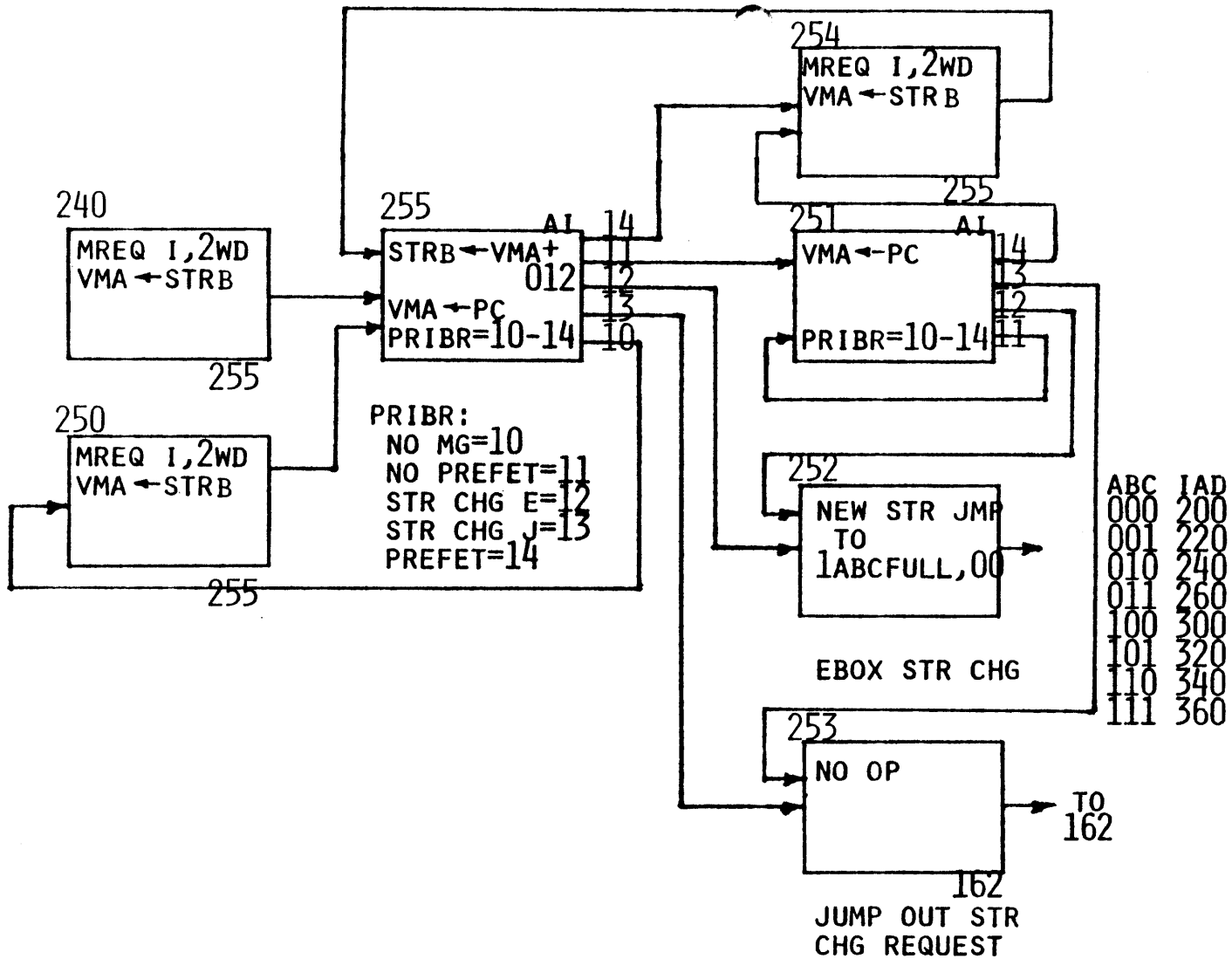
000= FLUSH PIPE  
 001= LD PC  
 002= CLEAR ERRS

DEC COMPANY CONFIDENTIAL

2080/KC10  
 IBOX PRESET, LOAD PC  
 D. HOOPER  
 13 AUG 80

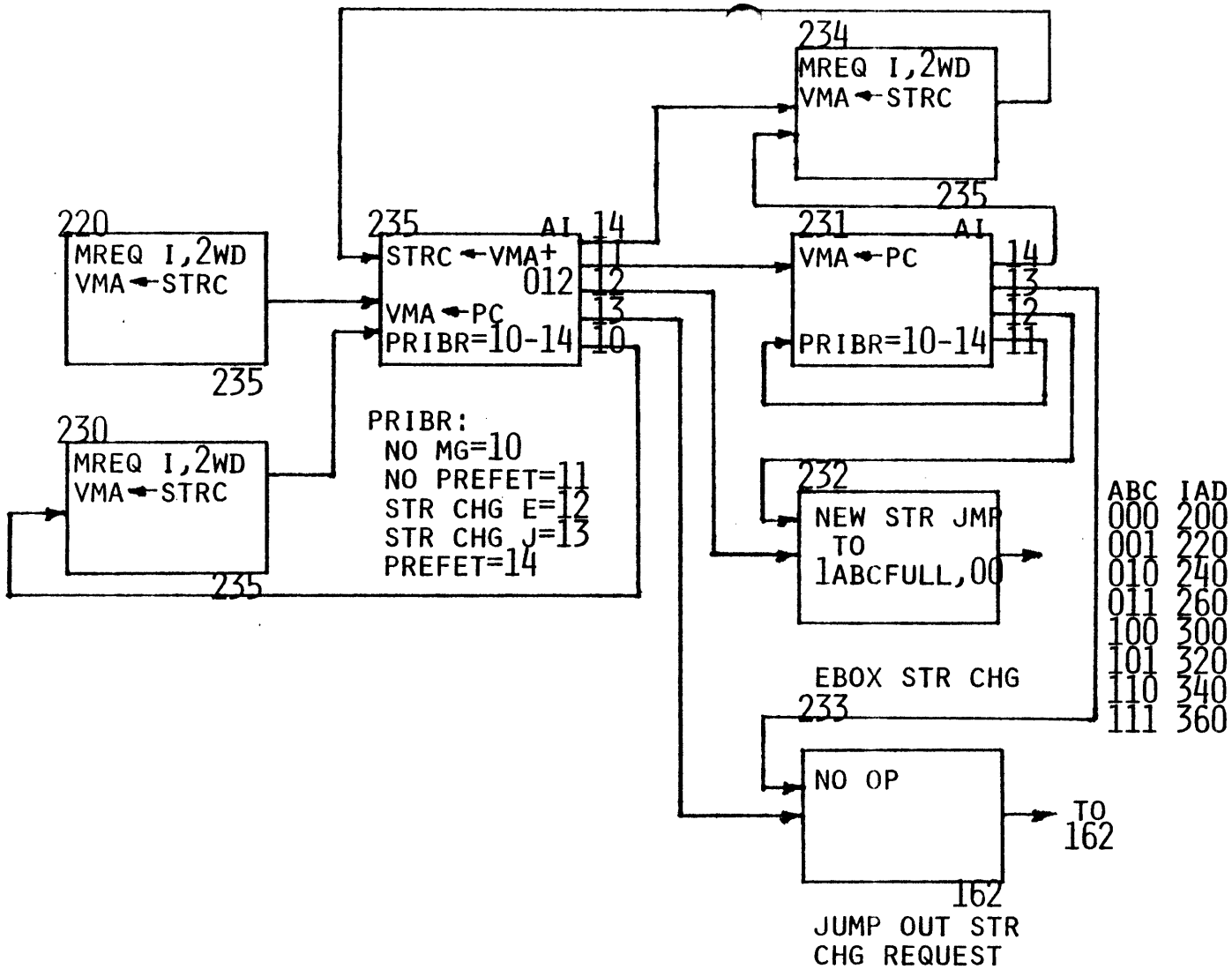


2080/KC10 IBOX  
STREAM A PREFETCH  
D. HOOPER  
14 AUG 80



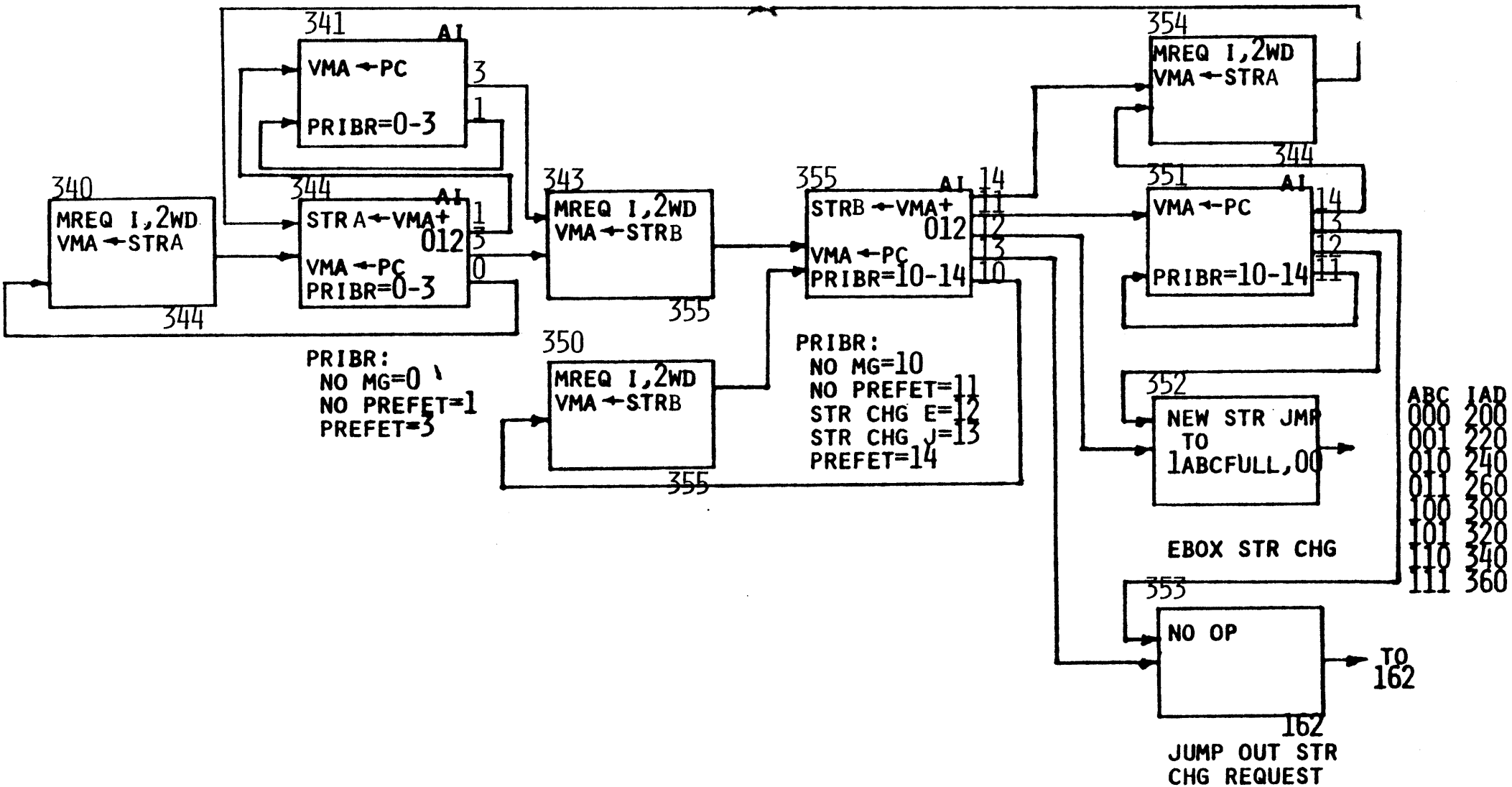
DEC COMPANY CONFIDENTIAL

2080/KC10 IBOX  
STREAM B PREFETCH  
D. HOOPER  
14 AUG 80



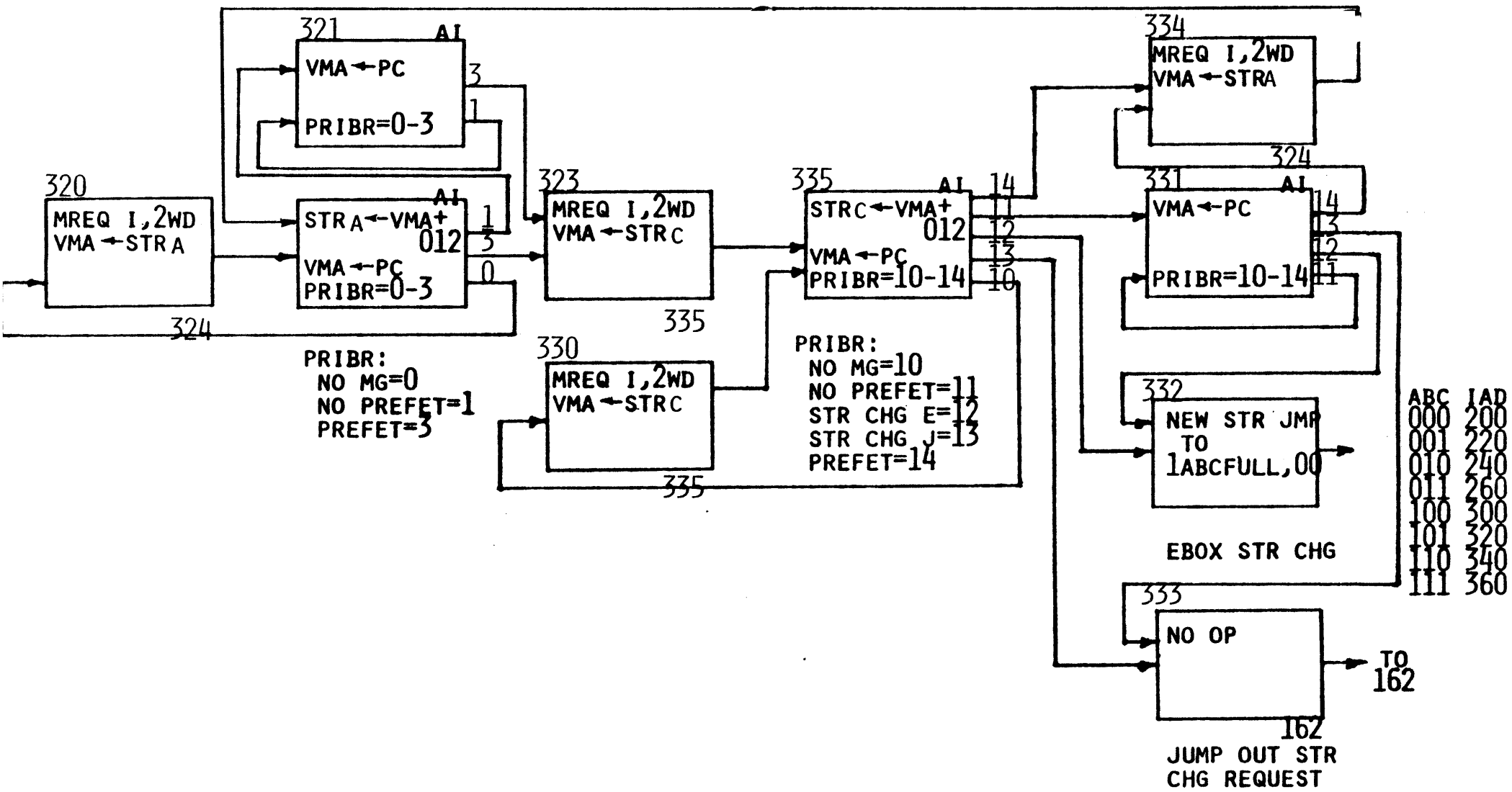
DEC COMPANY CONFIDENTIAL

2080/KC10 IBOX  
STREAM C PREFETCH  
D. HOOPER  
14AUG 80



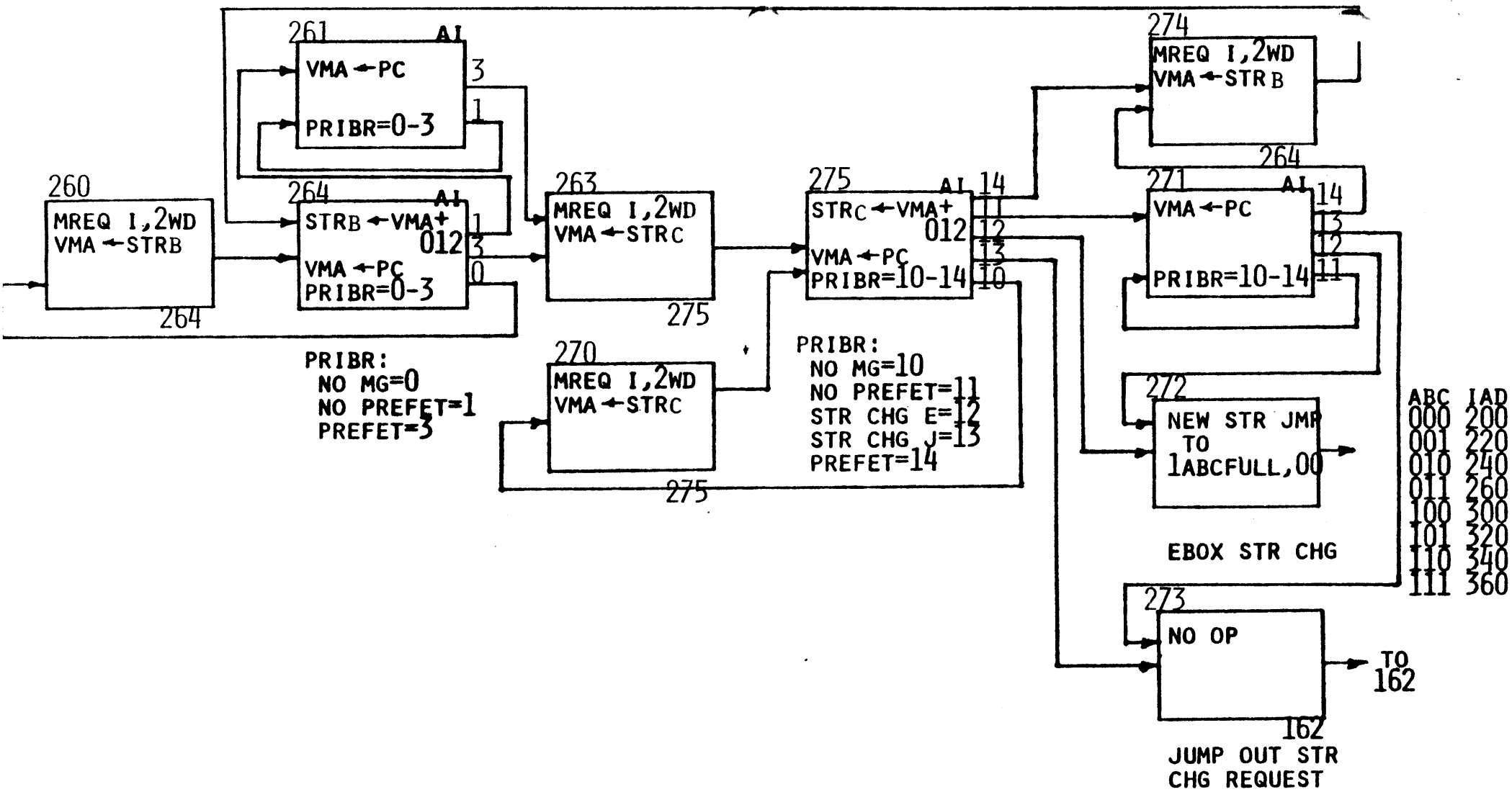
DEC COMPANY CONFIDENTIAL

2080/KC10 IBOX  
STREAM AB PREFETCH  
D. HOOPER  
14 AUG 80



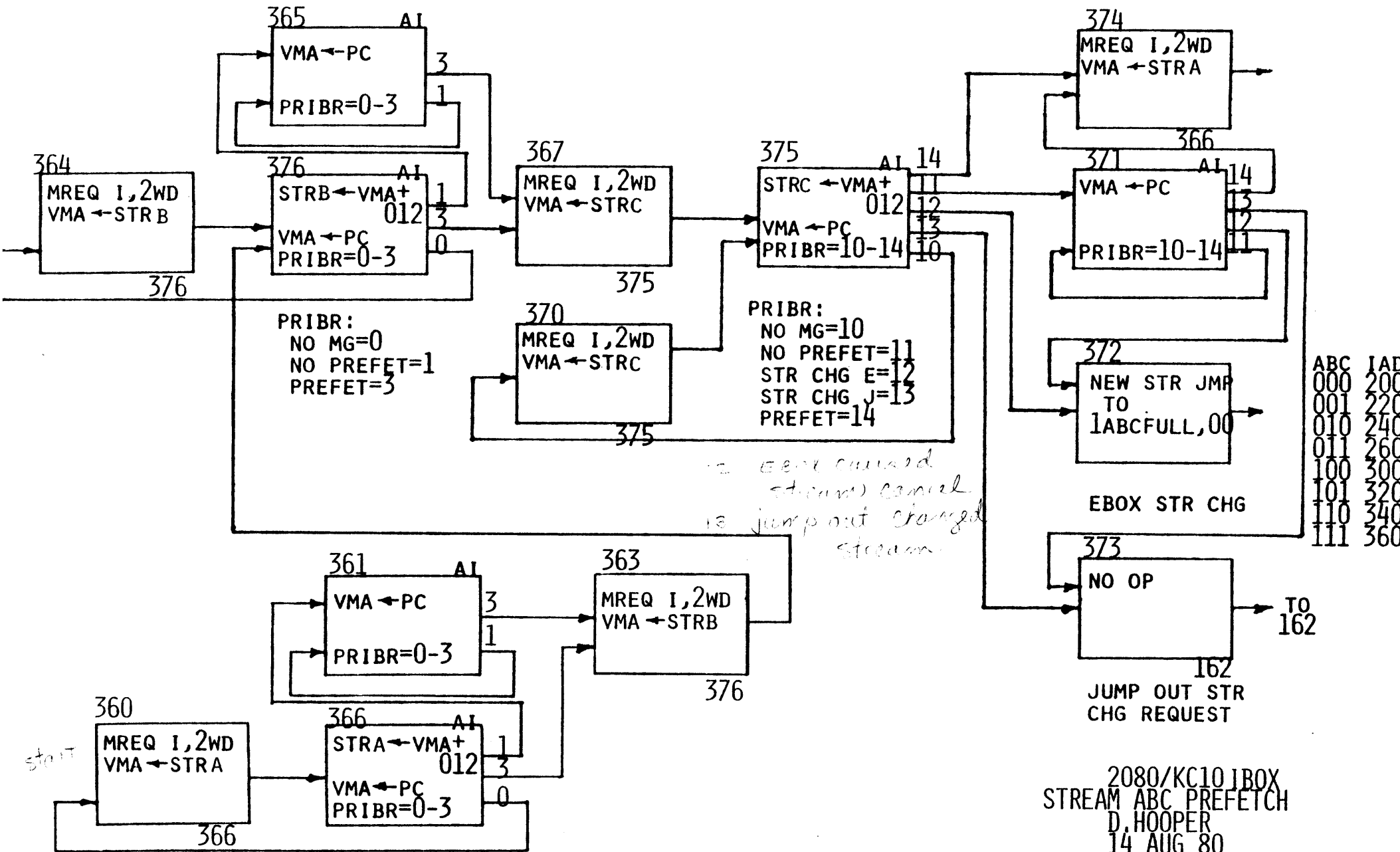
DEC COMPANY CONFIDENTIAL

2080/KC10 IBOX  
STREAM AC PREFETCH  
D. HOOPER  
14 AUG 80



DEC COMPANY CONFIDENTIAL

2080/KC10 IBOX  
STREAM BC PREFETCH  
D.HOOPER  
14AUG 80



PRIBR:  
 NO MG=0 no MGRANT → loop & wait for it  
 NO PREFET=1 no prefetch → loop & wait (every caught up w/ IBOX)  
 PREFET=3

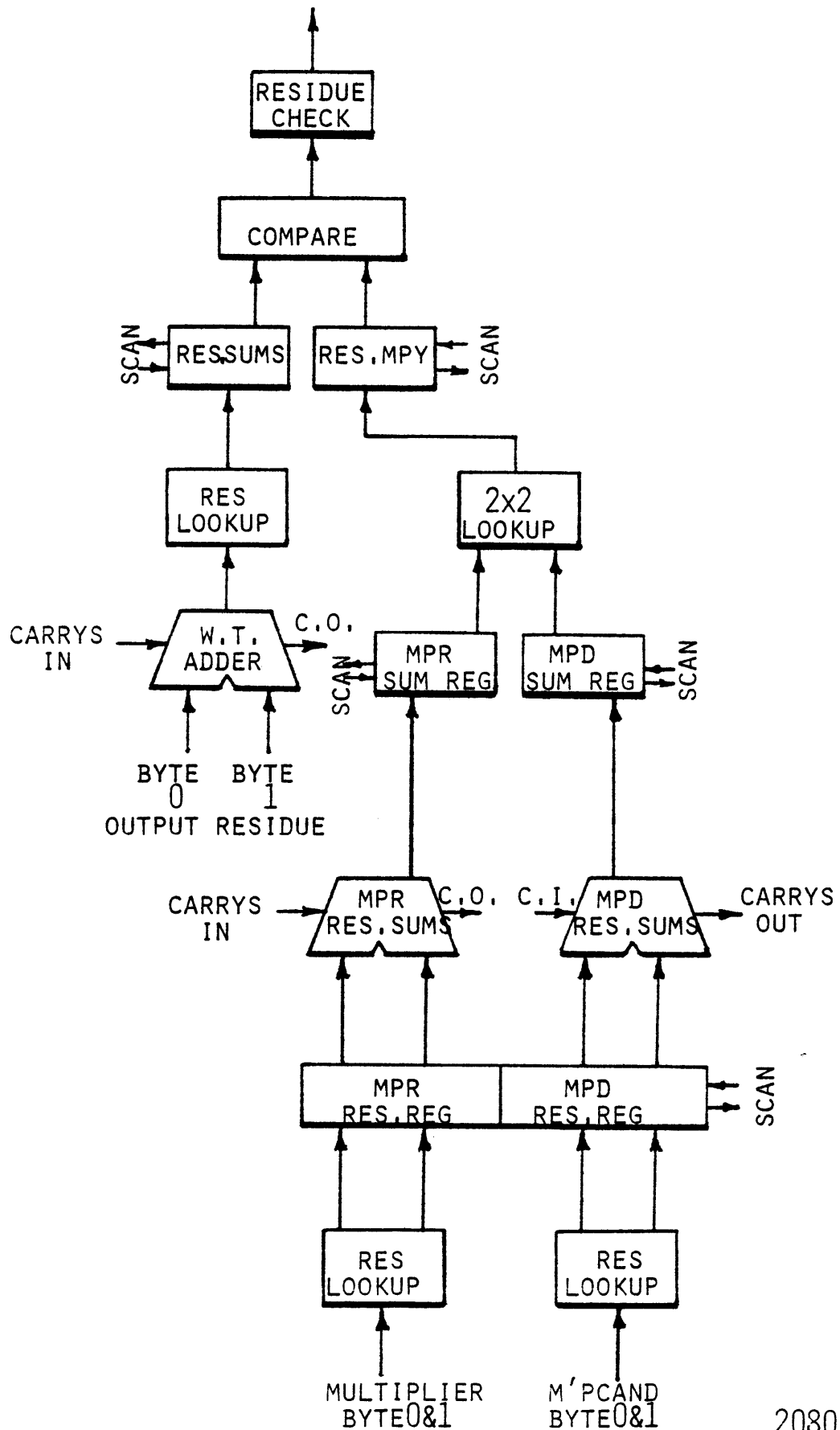
DEC COMPANY CONFIDENTIAL

2080/KC10 IBOX  
 STREAM ABC PREFETCH  
 D. HOOPER  
 14 AUG 80

STRB ← stream A  
 PRIBR ← MGRANT

FPA



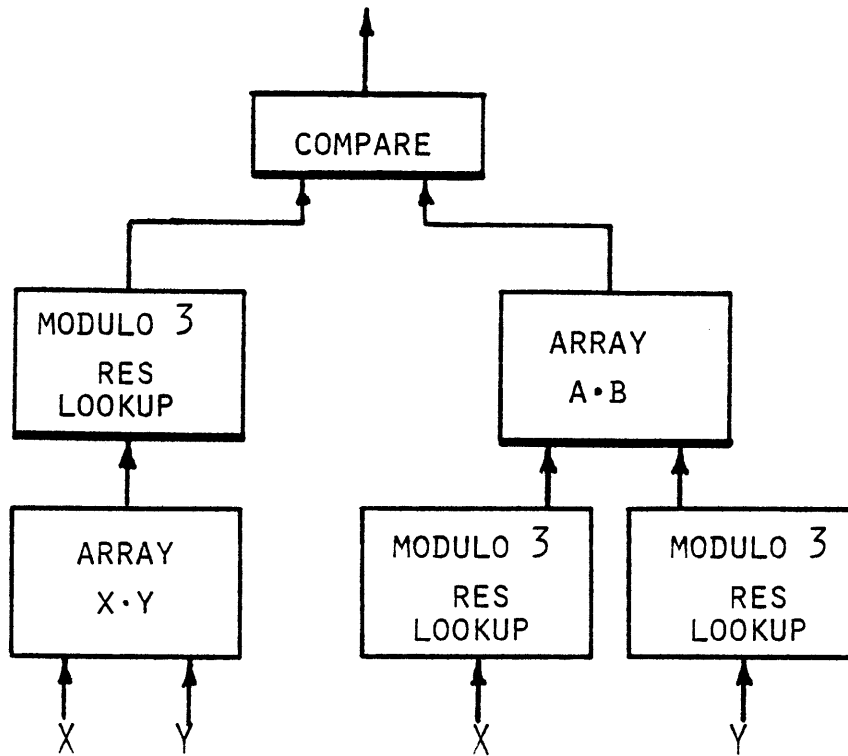


D. HOOPER  
3/5/80

DEC COMPANY CONFIDENTIAL

2080  
MODULO 3  
MULTIPLY  
ARRAY  
CHECKER

## 2080 MODULO 3 CHECKING ALGORITHM



EXAMPLE: X Y

$$X = (X-A) + A \quad \text{WHERE } A = \text{REMAINDER AFTER } \frac{X}{3}$$

$$Y = (Y-B) + B \quad \text{WHERE } B = \text{REMAINDER AFTER } \frac{Y}{3}$$

$$X \cdot Y = [(X-A) \cdot (Y-B)] + [A \cdot (Y-B)] + [B \cdot (X-A)] + [A \cdot B]$$

SINCE: RESIDUE  $(X-A) \equiv 0$

AND RESIDUE  $(Y-B) \equiv 0$ ,

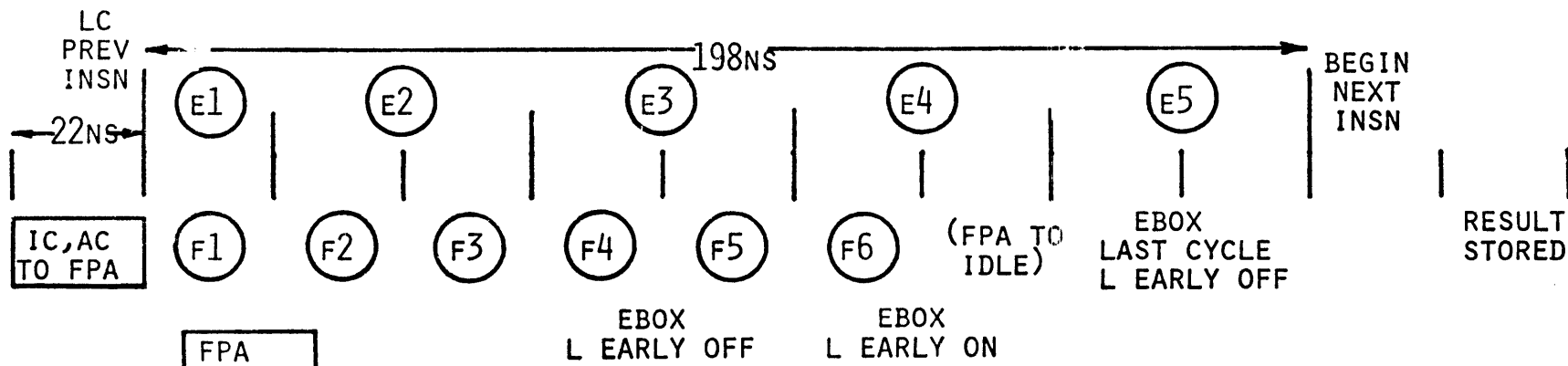
THEN: RESIDUE  $(X-A) \cdot (Y-B) \equiv 0$

RESIDUE  $A \cdot (Y-B) \equiv 0$

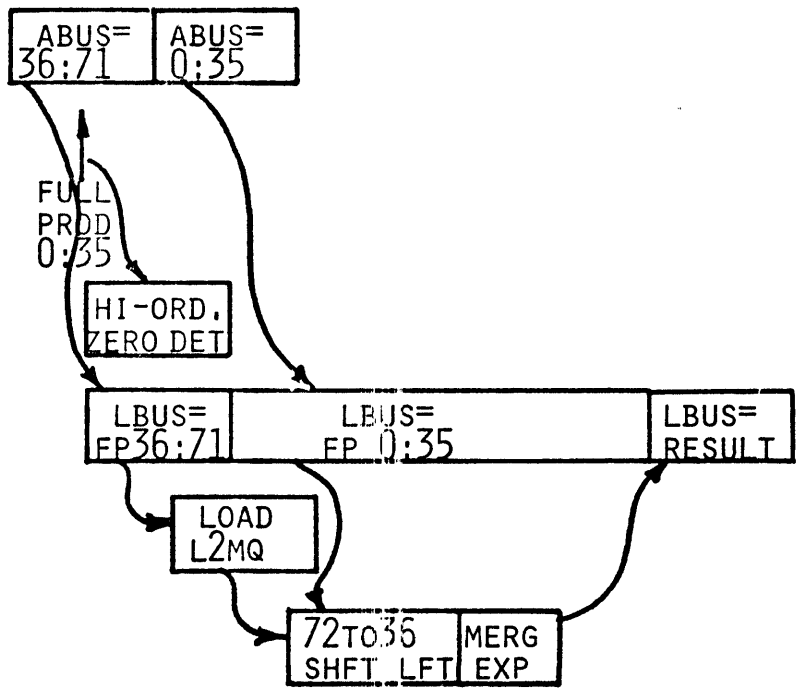
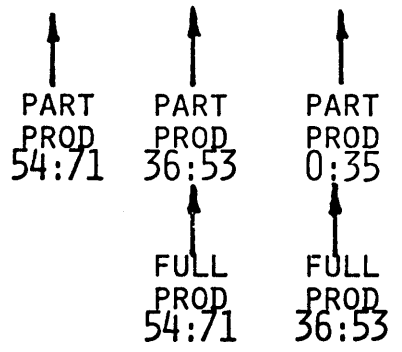
RESIDUE  $B \cdot (X-A) \equiv 0$

RESIDUE  $X \cdot Y \equiv A \cdot B$

DEC COMPANY CONFIDENTIAL



FPA UCODE  
(CTRL TO SETUP MPLR, MPCAND)



DEC COMPANY CONFIDENTIAL

*single precision  
Floating point copy  
in FPA*

KC10/2080  
FMP  
D. HOOPER  
19 AUG 80