DECUS
PROGRAM LIBRARY

DECUS NO.	8-343
TITLE	RADIAL INTERFACE INCLUDING INTERRUPT MASK FOR THE PDP-8 OR LINC-8
AUTHOR	Paul F. Sullivan
COMPANY	Cornell Aeronautical Laboratory, Inc. Buffalo, New York
DATE	April 10, 1970
SOURCE LANGUAGE	

RADIAL INTERFACE INCLUDING INTERRUPT MASK FOR THE PDP-8 OR LINC-8

ABSTRACT

This document describes a hardware modification to the PDP-8 or LINC-8 which protects software from obsolescence caused by the addition of new devices to the interrupt and/or data break facilities and allows significant savings of money and effort in interfacing further devices to the computer. The hardware also provides the computer with a dynamic priority interrupt facility.

1.0 INTRODUCTION

The environment of a real-time computer in a research laboratory is quite different from that of its process control or batch processing cousin. In the laboratory, peripherals change rapidly and unpredictably. These changes, especially those involving the interrupt or the data break facility, often make real-time software obsolete at an alarming rate. On the other hand, peripherals are more likely to be clustered around the computer in the laboratory than they are in the industrial plant. Clustered peripherals can be serviced more economically by a radial interface than by the standard DEC bus interface. This document describes an interface designed to provide maximum flexibility, economy, and system stability in the research laboratory environment. Sections 2.0 and 2.1 are concerned with software aspects of the interface and Sections 2.0, 2.2, 3.0, 4.0, and 5.0 are concerned with the hardware.

The interface (Fig. 1) consists of four main sections: interrupt mask, IOT generator, accumulator output, and accumulator input, each of which is described in detail below.

2.0 INTERRUPT MASK

In a laboratory situation, new devices are constantly being added to the computer facility. If these devices use the interrupt or the data break facility, software must be constantly updated to handle these events. Considerable time can be saved if the software can turn off all unwanted devices, even those it does not know about. It was to accomplish this that the interrupt mask was designed. (In the LINC-8, the LINC select flip flop provides similar masking for the LINC subsystem.)

In a system using the interrupt mask, device interrupt requests are ANDed with appropriate bits of a mask register before being fed to the computer's interrupt facility. Thus an interrupt occurs only if the mask bit corresponding to the device requesting the interrupt is set. Furthermore, in order that software written to handle only the initial interrupt mask be not obsoleted by the inclusion of additional masks, the output of each additional mask is ANDed with the most significant bit of the immediately previous mask and causes interrupts only through that mask. Wiring of a second 12-bit mask is shown in the sample interface. The same masking can also be applied to data break requests if a DM01 multiplexer is included in the system.

2.1 SOFTWARE ASPECTS

Each 12-bit interrupt mask in the interface is controlled by three IOT pulses. IOT 1 reads the mask register into the accumulator. (It does not clear the accumulator first.) IOT 2 clears the mask. IOT 4 sets all mask bits corresponding to bits set in the accumulator. Thus IOT 6 (clear and set) would usually be used for setting the mask to the accumulator bit pattern, while IOT 4 would be used to add more devices to those already selected. (We have chosen device codes 53 and 54 for the first and second interrupt masks, respectively.)

The manner of adding mask registers makes it possible for software which uses devices only from the first group of 11 to operate properly in any future hardware configuration provided it starts with the sequence

```
CLA
TAD   MASK
6536/Clear and Set the First Mask
```

where the most significant bit of MASK is 0 and only those bits of MASK are

set which correspond to devices the software recognizes. Similarly, software using the first two groups of devices should start with the sequence

```
        CLA
        TAD   MASK 1
        6536
        CLA
        TAD   MASK 2
        6546
```

where the most significant bit of MASK 1 is 1 and that of MASK 2 is 0.

The interrupt mask can also be used to effect a priority interrupt system. If the routine servicing a device masks out all devices of lower or equal priority, the interrupt facility can be turned on allowing higher priority devices to interrupt the service routine. The interrupt mask, of course, must be restored before exiting the service subroutine. (Note: to prevent havoc, the interrupt masks should be changed only with the interrupt facility turned off.) The following service subroutine allows higher priority interrupts:

```
SERV A,  DCA   AC       /save the accumulator
         TAD   0
         DCA   RETURN   /and the return address
         6533           /read and clear the interrupt mask
         DCA   MASK 1   /save it
         TAD   MASK A
         6534           /set for higher priority interrupts
         ION
          .
          .
          .
          .            /service device A
          .
          .
          .
          .
```

```
IOF
CLA
TAD   MASK 1
6536
CLA
TAD   AC            /restore the accumulator
ION
JMP   I    RETURN   /go back to interrupted program
```

A case of particular interest is that of the magnetic tape instructions on a LINC-8. The LINC tapes are accessed by PROGOFOP with the interrupt turned on. In order to avoid interfering with the tape transfers, PROGOFOP clears flags which arise during tape servicing without attending to the interrupting device. In the case of the Teletype paper tape reader, this results in the tape advancing without the character's being read into the computer. With the interrupt mask, all other interrupts can be masked out while the tape is being serviced. Since the computer does not see other flags, they need not be cleared, and devices such as the paper tape reader will continue to function properly. The following is a set of patches which mask out all non-LINC interrupts during tape servicing and restore the mask to its original condition when the tape routine is completed. The specific locations called out are appropriate for PROGOFOP Version II, (DEC-L8-SFAO-D) September 1, 1967.

Location	Contents	
1011	4132	JMS MASKCL
0111	4140	JMS MASKST
0132	0000	MASKCL, 0
0133	7200	CLA

```
0134    6533    6533          /read and clear the mask
0135    3137    DCA MASK      /save the old mask
0136    5532    JMP I MASKCL  /return
0137    0000    MASK, 0       /old mask
0140    0000    MASKST, 0
0141    7200    CLA
0142    1137    TAD MASK      /get the old mask
0143    6536    6536          /reset the mask
0144    7200    CLA
0145    5540    JMP I MASKST  /return
```

On the LINC-8, the following patch should be made to the LOAD program, Version II, (DEC-L8-L2AA-D) January 20, 1967.

```
0002    5151    JMP 151

0151    7200    CLA
0152    1156    TAD MASK
0153    6536    6536          /clear and set the mask
0154    7200    CLA
0155    5003    JMP 3
0156    0003    MASK, 0003    /teletype only
```

In the above patch, the MASK word should be changed to match the particular PROGOFOP on the tape if it is not the standard version. Since the interrupt masks are active registers vital to program operation, provision should be made for saving and restoring the masks in the KR01 power failure programming if the computer is suitably equipped.

2.2 HARDWARE ASPECTS

The interrupt mask consists of a flip flop register each bit of which drives one diode of a pair on an R141 gate (Fig. 2). The other diode of each pair is connected directly to the flag of a device to be masked. The output of the R141 drives the interrupt request line. Thus an interrupt

occurs if and only if a device flag is set and the interrupt mask bit corresponding to that device is set.

In the interface shown, four R203 flip flop modules have been used to drive two R141 gates providing a 12-bit mask. The extra two gates on the R141's can be used for any device which has multiple flags which need not be masked individually. In such a case, one flag is connected in the normal manner and the additional flags are each connected to one diode of an unused pair with the complementary diodes jumpered to the appropriate mask bit (Fig. 2). The most significant bit of the mask is reserved for ANDing with the output of the next inferior mask. This wiring is illustrated in the sample interface by the second interrupt mask. The outputs of the R141's of this second mask are wired together, connected to a spare load on the R111 at C32V, and feed the first mask through the spare inverter at C6U.

The data break request is masked by the same register as the interrupt request. The line driver in this case, however, is an R111 or an R113. One diode input of an R111 gate is connected to the break request flip flop in each device and the other diode is connected to the appropriate mask flip flop. The node point is connected through diodes to the most significant bit of any and all masks superior to the one controlling this device. Since this latter gating is not required on the first mask, an R113 can be used instead of the R111. The output of each gate drives one of the DM01 break request input lines.

3.0 IOT GENERATOR

The IOT generator (Fig. 3) is prewired to handle 26 different devices. The buffered MB lines for bits 3 through 8 for both the 0 and the 1 state are fed to inverters (R107) which will supply sufficient current to drive all the device selectors while drawing only 1 ma from each BMB line. The inverters

also make it possible to code the device selectors in the standard way, i.e. by removing diodes driven by the nonasserted lines, rather than in complementary fashion as is required when the BMB lines are used directly. The IOP pulses are buffered by two series inverters each and the resulting negative pulses are bussed down the row of device selectors. The standard device selector (W103) may be used in the interface, but additional flexibility can be had at a lower cost by using a B171 module in the lower row of the device selector and an R111 in the upper row. (Pulse amplifiers may be added where needed in the device electronics. In the interrupt mask, only IOT 4 requires a PA for a saving of $9 over a W103 selector.) Pins E and F of the lower row and pins E, L, and S of the upper row are jumpered together to provide the appropriate interconnections for this module pair. These jumpers should be removed if W103 modules are used.

A case in which this added flexibility is advantageous is illustrated in the leftmost 16 module locations within the IOT generator. In about 60 percent of the peripherals listed in the PDP-8 handbook, IOT 1 is used as a "skip on flag" instruction. By connecting the output from the B171 through a diode to the node point of the IOP 1 gate on the R111 (pin T) rather than jumpering it directly to pin S, an additional condition for pulse generation is created. If the device flag is now connected to pin S of the R111, and the output, without a clamped load, is connected to the skip bus, the "skip on flag" instruction is implemented. In the interface, 16 module locations have been prewired for this mode of operation. To implement a skip on flag, one has only to insert the two modules and connect the device flag to pin S of the R111. (W103 device selectors should not be used in these locations.)

The row of modules containing the R111's also has a positive going power clear pulse available from 15 different inverters. If flip flops within a

device are to be cleared by an IOT and by a power clear, it is necessary only to jumper the IOT output (without a clamped load) to the output of an unused power clear inverter (cf. Fig. 4). This gives a hard wire ORed clear pulse of 18 ma, sufficient to clear 18 flip flops. Negative going power clear pulses are also available at three locations.

4.0 ACCUMULATOR OUTPUT

The accumulator output section is merely additional buffering on the BAC lines. Negative levels are provided at a capacity of 16 ma in the lower row of the output; positive levels at 18 ma are provided in both the lower and the upper rows.

5.0 ACCUMULATOR INPUT

Rather than the normal R123 bus driver, the accumulator lines are driven by R141 and/or gates for a savings of about $1.30 per device-bit. One diode of each diode pair is bussed along the entire row of 12 drivers (Fig. 5) and is connected to the output of an R107 inverter at the least significant bit end and to ground at the most significant bit end. The ground connection is necessary to prevent the reading of unwanted bits. If a device is to input information into the low order n bits of the accumulator on a given IOT, connect the clamped load to the required IOT gate of the R111 in the selector section, connect this output to the input of an unused inverter on the R107 of the accumulator input section, cut the bus driven by this inverter to the left of the nth bit, and connect the device bit lines to the diodes paired with those now driven by the inverter. This procedure was followed in wiring the IOT 1 instruction for the interrupt mask. (Since the second interrupt mask has been prewired but not implemented, the ground on that bus has been left intact.)

Additional R141 banks may be paralleled to the original banks to provide for more devices. In the interface shown, one additional bank has been wired so that 14 input devices may be handled with low order bit transfers. The circuitry can be used more efficiently if some devices transfer into the high order bits unused by other devices. A slot for an additional R107 is provided at the left end of the upper R141 bank for this purpose.

FIGURE CAPTIONS

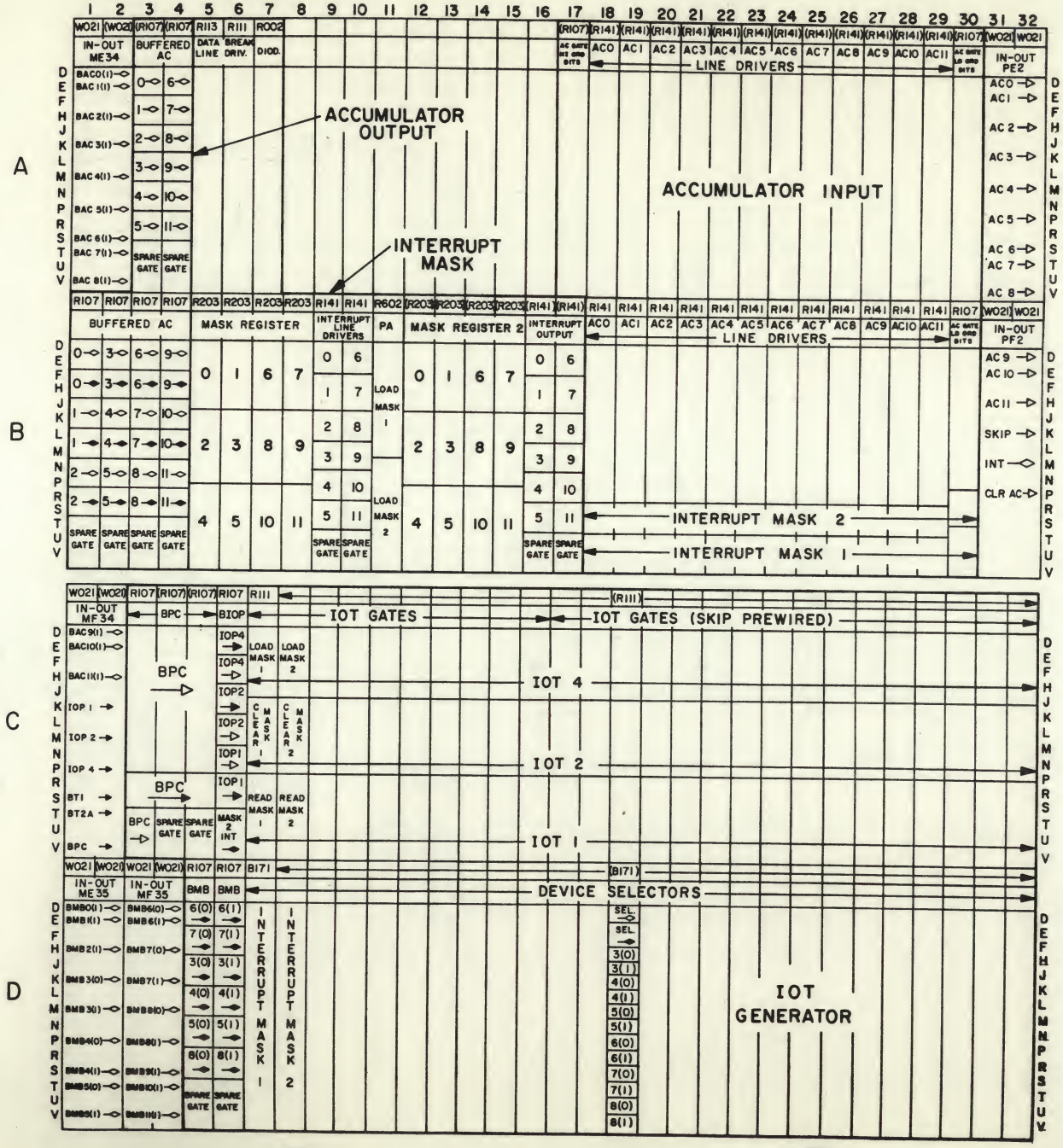
Figure 1: Module Assignment

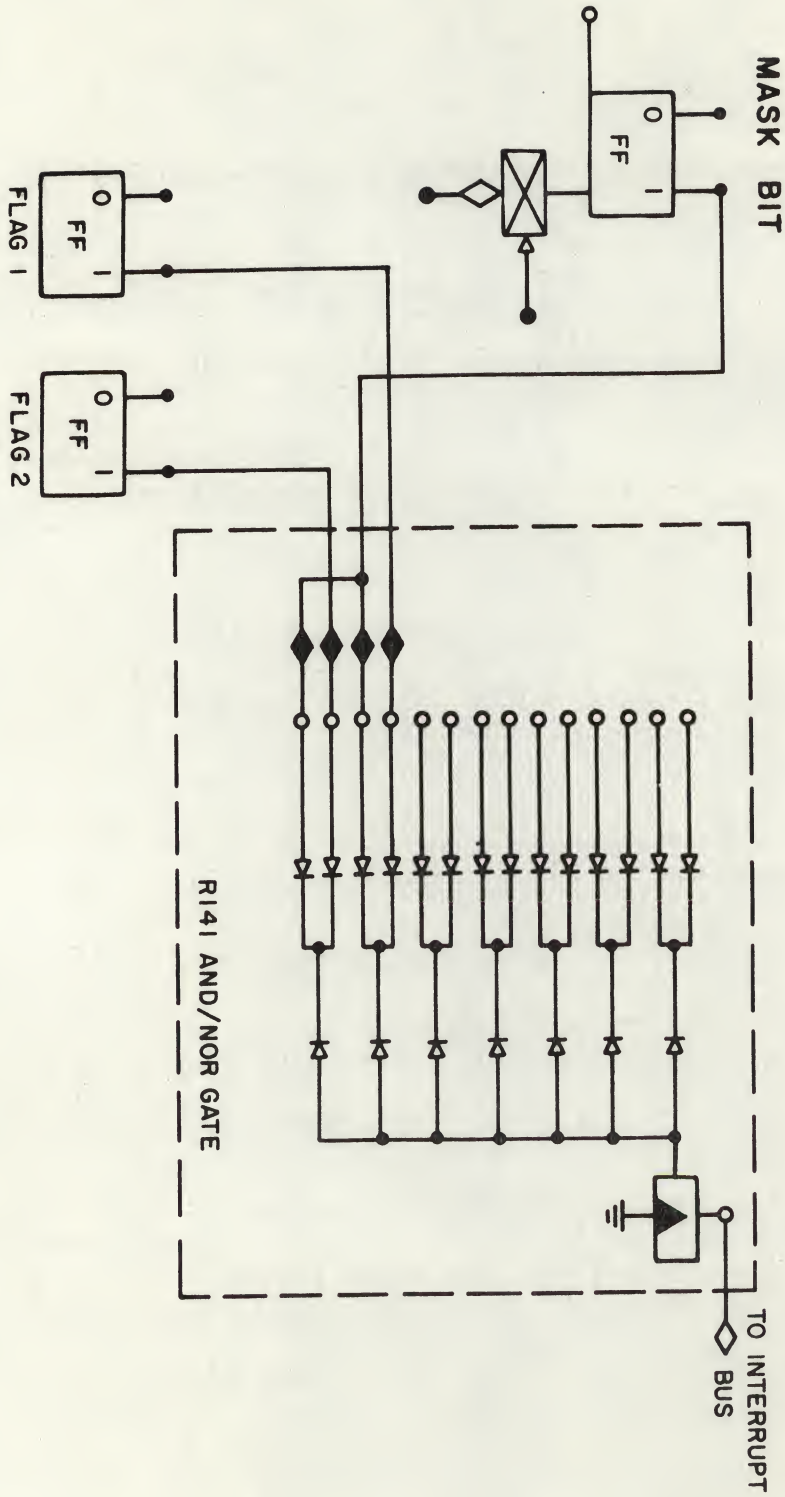
Figure 2: Connecting a Device with Multiple Flags to the Interrupt Mask

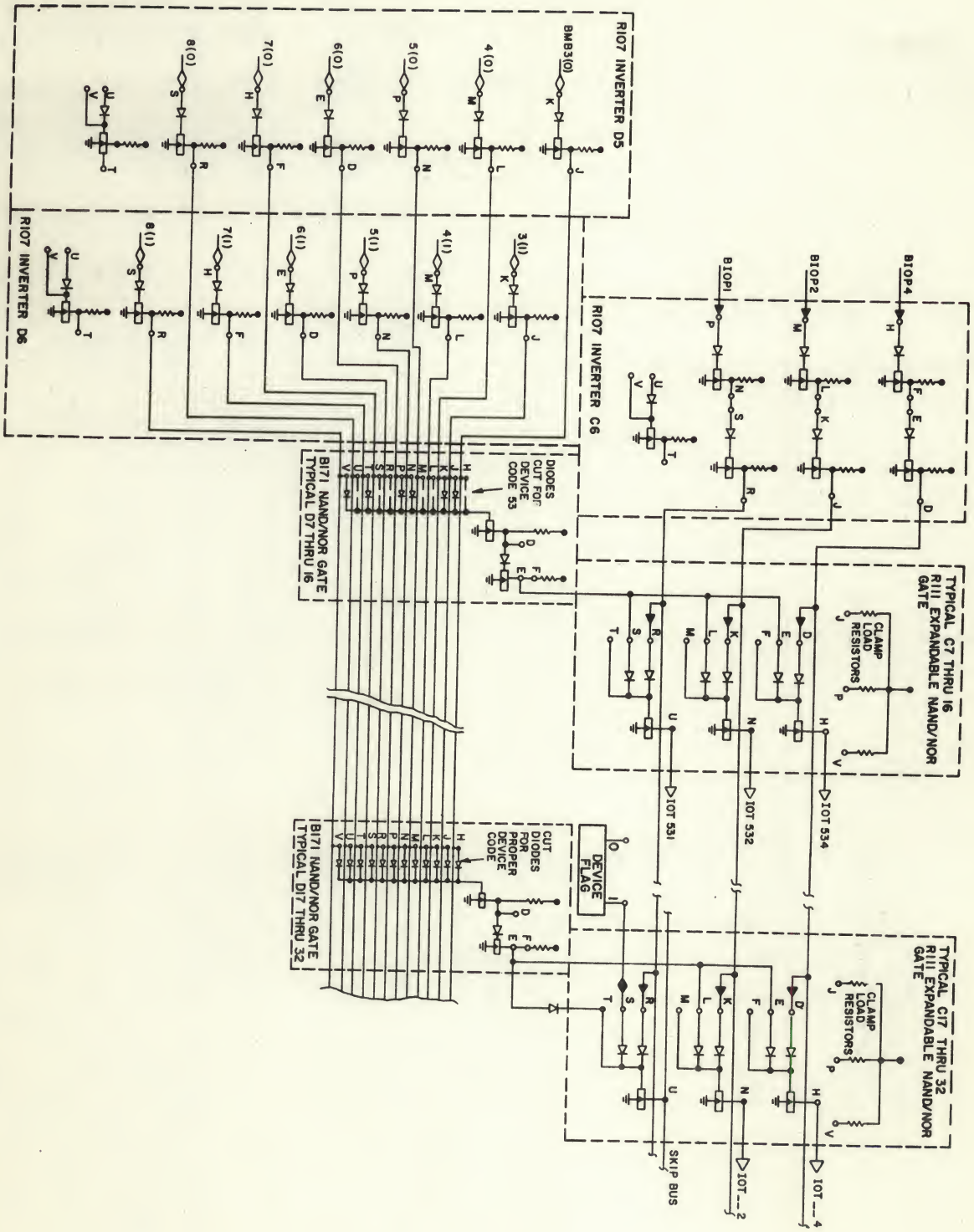
Figure 3: IOT Generator

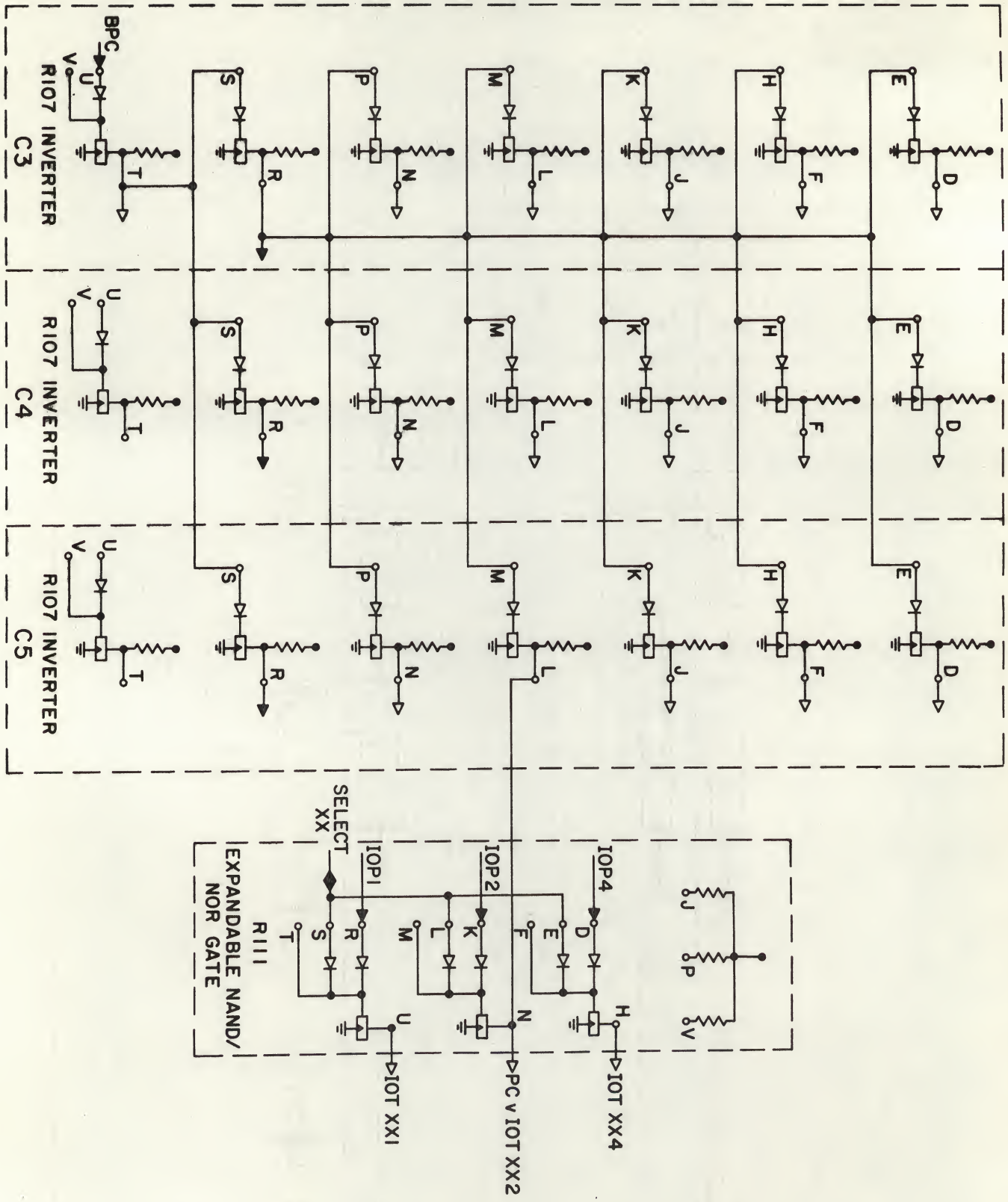
Figure 4: ORing an IOT with a Power Clear

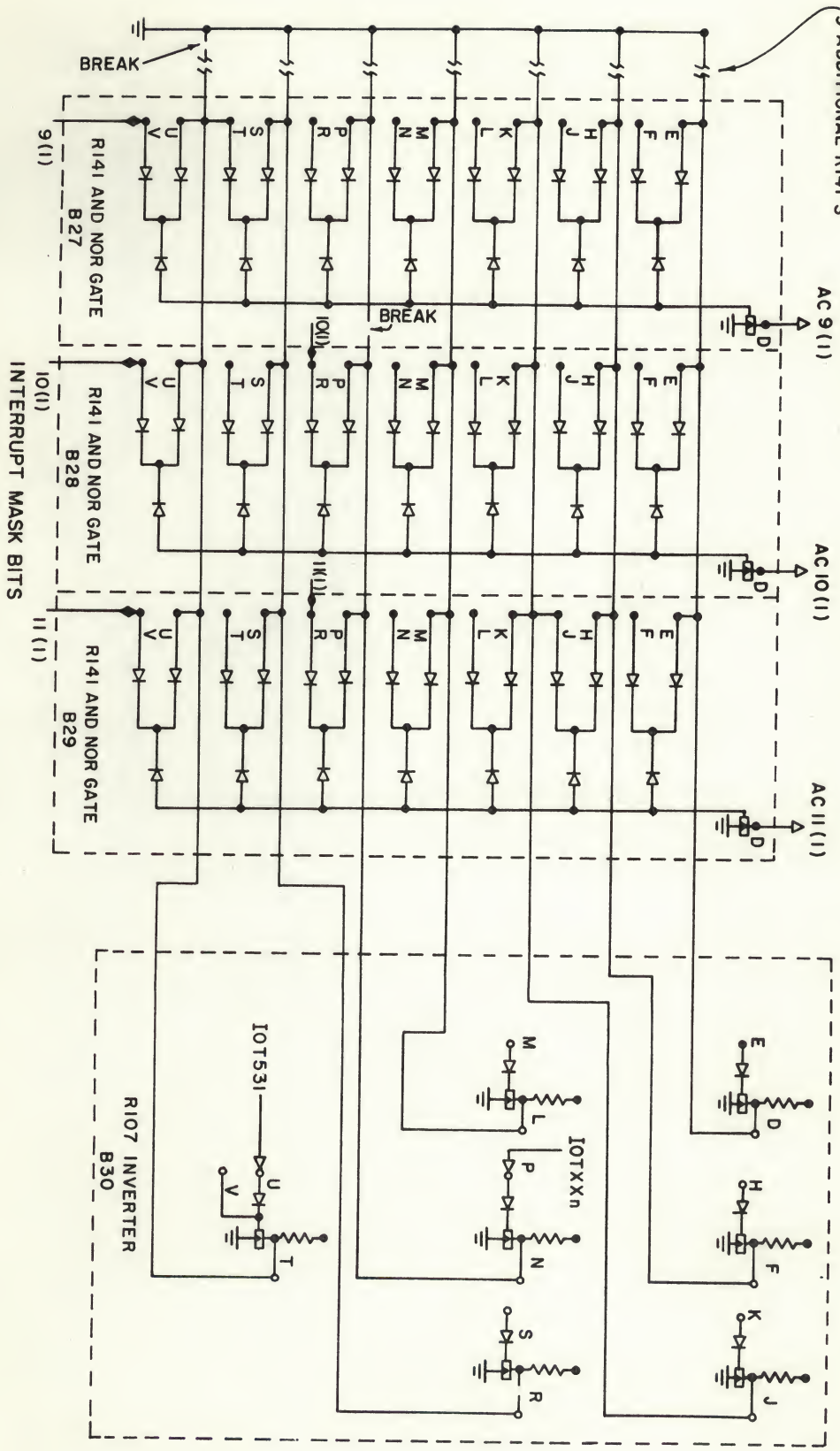
Figure 5: Accumulator Input
(Device XX shown transferring two bits only on IOT n)


ACCUMULATOR OUTPUT
INTERRUPT MASK
ACCUMULATOR INPUT
LINE DRIVERS
INTERRUPT MASK 2
INTERRUPT MASK 1
IOT GATES
IOT GATES (SKIP PREWIRED)
IOT 4
IOT 2
IOT 1
DEVICE SELECTORS
IOT GENERATOR
BPC


MASK BIT
FF
O
I
FLAG 1
FLAG 2
R141 AND/NOR GATE
TO INTERRUPT BUS


RIO7 INVERTER D5
RIO7 INVERTER D6
RIO7 INVERTER C6
BM83(0)
4(0)
5(0)
6(0)
7(0)
8(0)
3(1)
4(1)
5(1)
6(1)
7(1)
8(1)
BIOP1
BIOP2
BIOP4
TYPICAL C7 THRU I6 RIII EXPANDABLE NAND/NOR GATE
TYPICAL C17 THRU 32 RIII EXPANDABLE NAND/NOR GATE
CLAMP LOAD RESISTORS
DIODES CUT FOR DEVICE CODE 53
BI71 NAND/NOR GATE TYPICAL D7 THRU I6
CUT DIODES FOR PROPER DEVICE CODE
BI71 NAND/NOR GATE TYPICAL DI7 THRU 32
DEVICE FLAG
IOT 531
IOT 532
IOT 534
SKIP BUS
IOT --2
IOT --4


BPC
R107 INVERTER
C3
R107 INVERTER
C4
R107 INVERTER
C5
SELECT XX
IOP1
IOP2
IOP4
R111
EXPANDABLE NAND/ NOR GATE
IOT XX1
PC v IOT XX2
IOT XX4


9 ADDITIONAL R141'S
BREAK
9(1)
R141 AND NOR GATE B27
BREAK
IO(1)
R141 AND NOR GATE B28
IK(1)
II(1)
R141 AND NOR GATE B29
INTERRUPT MASK BITS
AC 9(1)
AC 10(1)
AC 11(1)
R107 INVERTER B30
IOT531
IOTXXn