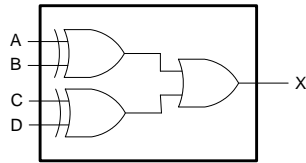


46h	4Ah	64h	6Eh	32h	35h
F	J	d	n	2	5

Se pretende rediseñar el circuito de la figura adjunta utilizando un array lógico programable (PLA).



- ¿Cuál es el número mínimo de puertas AND y OR debe tener dicho array lógico programable?

AND: 12 OR: 1

Se ha desarrollado el siguiente programa para la CPU teórica. El código fuente está compuesto de las siguientes partes: un **programa principal**, una **rutina de interrupción** correspondiente al periférico teclado y un **procedimiento llamado escribe_nota**.

El **programa principal** instala la rutina de teclado.

La **rutina de interrupción** comprueba si hay una tecla pulsada, y en caso de existir llamará al procedimiento `escribe_nota` pasando como parámetro la información que contiene el registro de datos del teclado.

El **procedimiento escribe_nota** recibe como parámetro el contenido del registro de datos de teclado con la información de la tecla pulsada. Si la tecla pulsada no es un número el procedimiento no imprime nada en pantalla. Por lo contrario si la tecla pulsada es un número el procedimiento escribirá en la pantalla la cadena

de texto correspondiente a dicho número que se encuentran definidas en la zona de datos. Cada una de las cadenas de texto finaliza con el terminador 0.

```
ORIGEN 1200h
INICIO main
.PILA ¿?
.DATOS
notas VALOR "cero",0, "uno",0, "dos",0,
"tres",0, "cuatro",0, "cinco",0, "seis",0,
"siete",0, "ocho",0, "nueve",0

.CODIGO
main:

---- HUECO 1 ----

MOVL R1, 26
MOVH R1, 0
MOV [R1], R0
STI
JMP -1

PROCEDIMIENTO rutina_teclado
PUSH R0
PUSH R1

---- HUECO 2 ----

MOVL R1, 1h ; R1=reg. control tecl.
MOVH R1, 60h ;

---- HUECO 3 ---- ; R2=mascara tecla
; pulsada

bucle_rutina:
MOV R3, [R1]
AND R2, R3, R2
BRZ no_hay_tecla

DEC R1 ; R1=reg. datos
MOV R3, [R1]
INC R1 ; R1=reg. control

PUSH R3 ; Parametro del proced.

---- HUECO 4 ----

JMP bucle_rutina
```

```
no_hay_tecla:
POP R4
POP R3
POP R2
---- HUECO 5 ----
FINP
```

```
PROCEDIMIENTO escribe_nota
PUSH R6
MOV R6, R7
PUSH R0
PUSH R1
PUSH R2
PUSH R3
```

```
---- HUECO 6 ----; R0 <- Recuperar parámetro
MOVH R0, 0h

MOVL R1, 39h
MOVH R1, 0h
COMP R1, R0
BRC no_es_numero

MOVL R1, 30h
COMP R0, R1
BRC no_es_numero
```

```
SUB R0, R0, R1
XOR R2, R2, R2
MOVH R1, BYTEALTO DIRECCION notas
MOVL R1, BYTEBAJO DIRECCION notas
```

```
bucle_proc:
COMP R0, R2
BRZ escribe
```

```
busca_final_nota:
---- HUECO 7 ----
COMP R3, R2
BRNZ busca_final_nota
DEC R0
JMP bucle_proc
```

```
escribe:
MOVL R0, 80h
MOVH R0, 60h
```

```
siguiente:
MOV R3, [R1]
COMP R3, R2
BRZ acaba
MOVH R3, 7h
MOV [R0], R3
```

INC R1