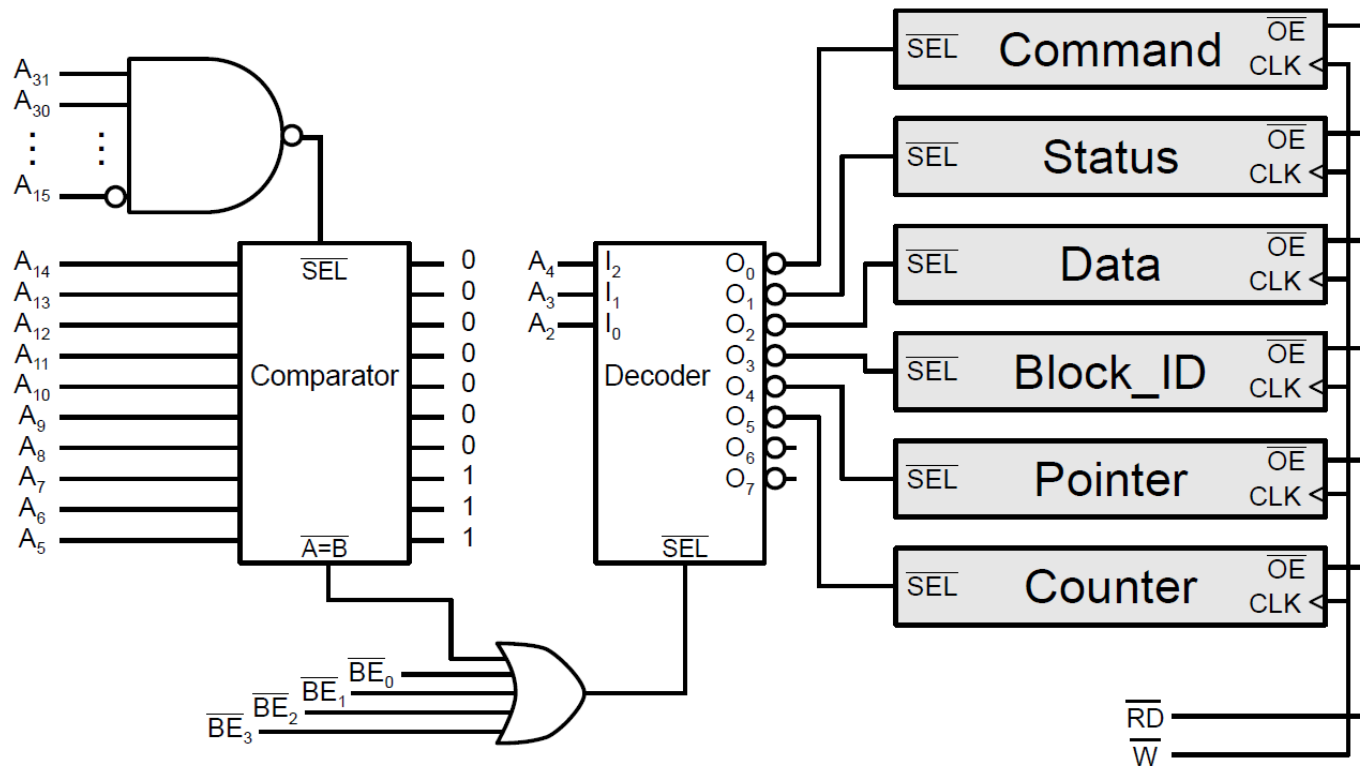


Ejemplo interfaz (nº7)



- Dirección Base= ¿?
- Número de puertos (direcciones) del interfaz= ¿?
- Direcciones de puerto de todos los registros
- Operaciones de lectura/escritura sobre los registros
- Número de bytes que ocupa el interfaz en el espacio de memoria= ¿?

Ejemplo interfaz (nº7)

$A_{31} A_{30} A_{29} A_{28} A_{27} A_{26} A_{25} A_{24} A_{23} A_{22} A_{21} A_{20} A_{19} A_{18} A_{17} A_{16} A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$

1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 0 0 0 0 | 0 0 0 0 | 1 1 1 0 | 0 0 0 0

0x F F F F 0 0 E 0

Registro	Dir. Puerto	Operación	Instrucción*
Command	DB	Lectura/Escritura	lw/sw
Status	DB+4	Lectura/Escritura	lw/sw
Data	DB+8	Lectura/Escritura	lw/sw
Block_ID	DB+12	Lectura/Escritura	lw/sw
Pointer	DB+16	Lectura/Escritura	lw/sw
Counter	DB+20	Lectura/Escritura	lw/sw

Teniendo en cuenta las líneas que intervienen en la selección del interfaz ($A_{31} \dots A_5$), quedan sólo 5 líneas ($A_4 \dots A_0$) para direccionamiento interno del interfaz, por lo que este ocupará $2^5 = 32B$ del espacio direccionable, esto es, espacio que no estará disponible para otros interfaces o módulos de memoria

- Programación -

```
        .data 0x10000000  
mi_dato: .word 0xFF000000
```

#almacena un '0' en DATA

```
la $t0, 0xFFFF00E0  
sw $0, 8($t0)
```

#almacena el contenido de DATA

#en mi_dato

```
la $t0, 0xFFFF00E0  
lw $t1, 8($t0)  
sw $t1, mi_dato
```

#pone a '1' los bits 0 y 7 de COMMAND

#sin modificar el resto

```
la $t0, 0xFFFF00E0  
lw $t1, 0($t0)  
ori $t1,$t1,0x81  
sw $t1, 0($t0)
```

Ejemplo interfaz (nº7)

#pone a '0' el bit 3 de COMMAND

#sin modificar el resto

```
la $t0, 0xFFFF00E0  
lw $t1, 0($t0)  
andi $t1,$t1,0xFFF7  
sw $t1, 0($t0)
```

#si bit 7 de STATUS es '1' entonces

#almacenar mi_dato en DATA

#si no, almacenar '0'

```
la $t0, 0xFFFF00E0  
lw $t1, 4($t0)  
andi $t1,$t1,0x80  
beqz $t1,salto  
lw $t1, mi_dato  
sw $t1, 8($t0)  
j exit
```

```
salto: sw $0, 8($t0)
```