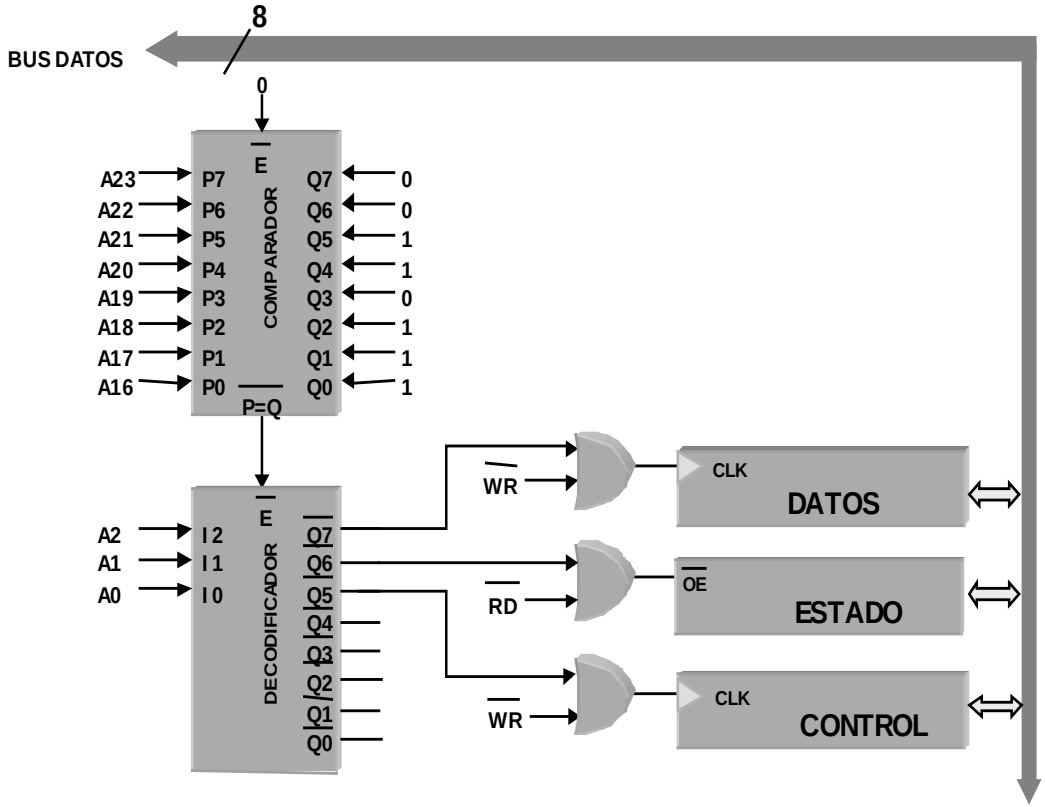
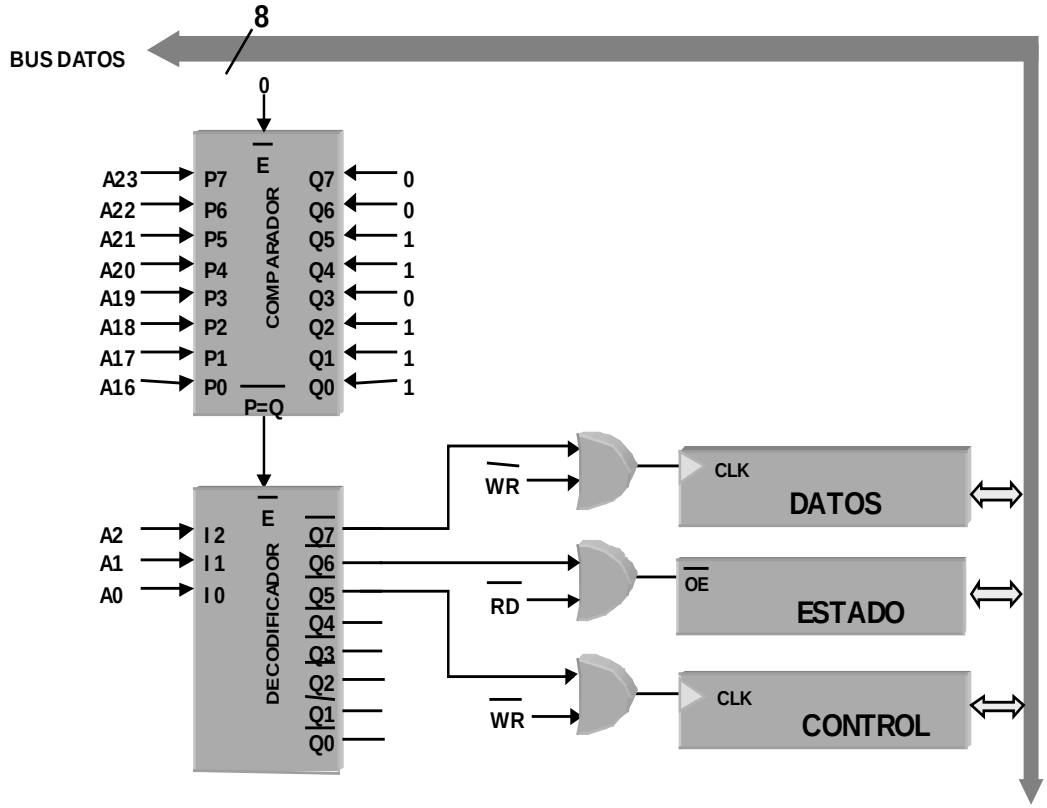
Ejemplo interfaz (nº3)



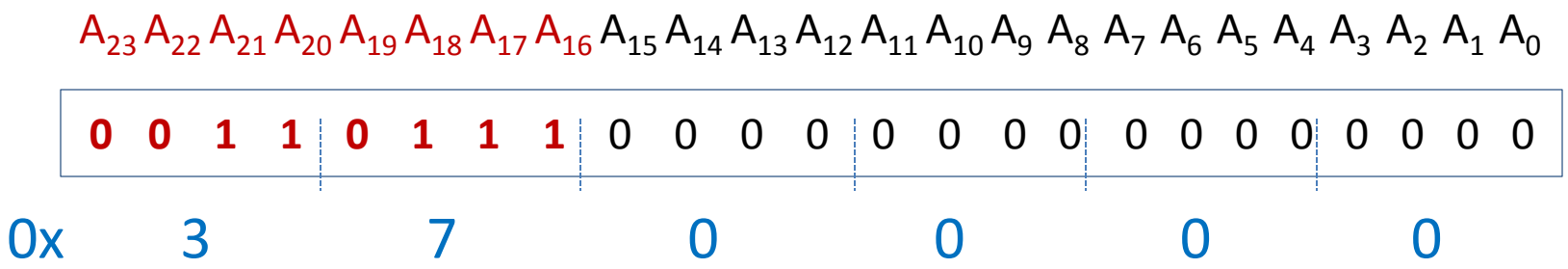
- Dirección Base= ¿?
- Número de puertos (direcciones) del interfaz= ¿?
- Direcciones de puerto de los registros DATOS, ESTADO y CONTROL
- Operaciones de lectura/escritura sobre los registros
- Número de bytes que ocupa el interfaz en el espacio de memoria= ¿?

Ejemplo interfaz (nº3)



- Dirección Base= **0x370000**
- Número de puertos (direcciones) del interfaz= **3**
- Direcciones de puerto de los registros DATOS (**BASE+7**), ESTADO (**BASE+6**) y CONTROL (**BASE+5**)
- Operaciones de lectura/escritura sobre los registros: DATOS (**escritura**); ESTADO (**lectura**) y CONTROL (**escritura**)
- Número de bytes que ocupa el interfaz en el espacio de memoria= **2^{16}= 64KB**

Ejemplo interfaz (nº3)



Registro	Dir. Puerto	Operación	Instrucción*
DATOS	DB+7	Escritura	sb
ESTADO	DB+6	Lectura	lb
CONTROL	DB+5	Escritura	sb

Hay sólo tres direcciones de Puerto disponibles en el interfaz, aunque podrían haber, de acuerdo a la lógica de decodificación, hasta 8 direcciones de puerto

Teniendo en cuenta las líneas que intervienen en la selección del interfaz (A23 …A16), éste ocupa 2^{16}= 64KB del espacio direccionable, esto es, espacio que no estará disponible para otros interfaces o módulos de memoria

(*) Aunque se trata de una CPU de 8 bits, se asume un juego de instrucciones tipo MIPS, pero limitada a instrucciones de byte

Ejemplo interfaz (nº3)

- Programación -

```
            .data 0x10000000
mi_dato: .byte  0x1A

#almacena un '0' en DATOS
la $t0, 0x370000
sb $0, 7($t0)

#almacena mi_dato en DATOS
la $t0, 0x370000
lb $t1,mi_dato
sb $t1, 7($t0)

#pone a '1' el bit 3 de CONTROL
la $t0, 0x370000
li $t1,0x08
sb $t1, 5($t0)
```

```
#almacena mi_dato en DATOS
#si bit 5 de ESTADO es '1'
#si no, almacenar 0
        la $t0, 0x370000
        lb $t1, 6($t0)
        andi $t1,$t1,0x20
        beqz $t1,salto
        lb $t1,mi_dato
        sb $t1, 7($t0)
        j exit
salto:  sb $0,7($t0)
```