



Laboratorio 1:

Acceso elemental a dispositivos (i)

dispositivos externos mapeados en memoria

Programación de sistemas y dispositivos

José Manuel Mendías Cuadros

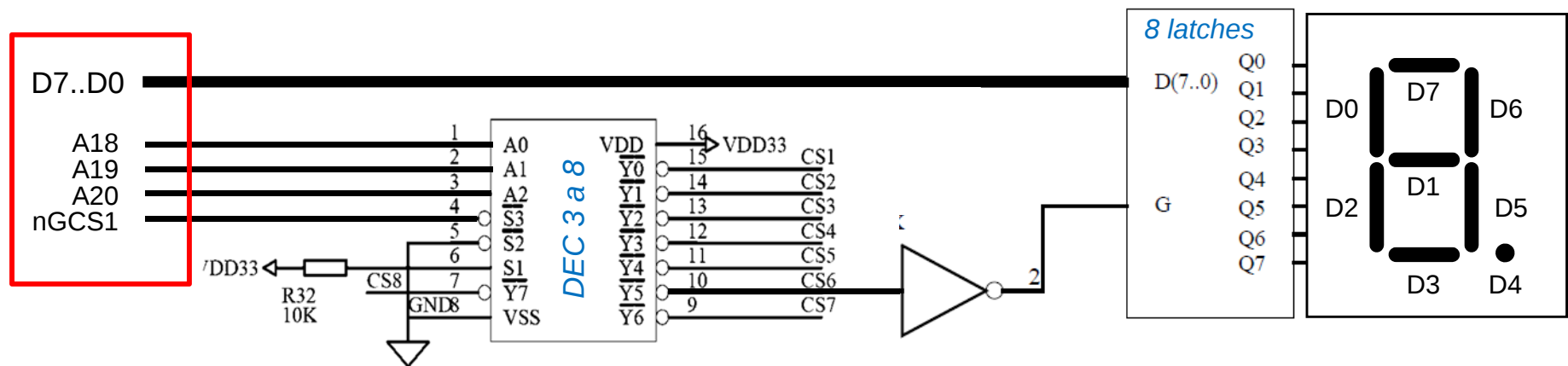
*Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid*



Conexión del display



- Al **bus de memoria** del SoC hay conectado un **display 7-segmentos**:



- El array de latches que estimulan los segmentos (**lógica inversa**) podrá cargarse **escribiendo un byte en cualquier dirección** del rango:

001X.XXX1.01XX.XXXX.XXXX.XXXX.XXXX
nGCS1 CS6

- Aprovecharemos que **la configuración básica del SoC está hecha** por el programa residente en ROM que se ejecuta tras reset.
 - o Pilas, interrupciones, memoria, de puertos de E/S, de reloj, etc...
 - o Si no existiese tal programa, la aplicación debería configurarlo todo.



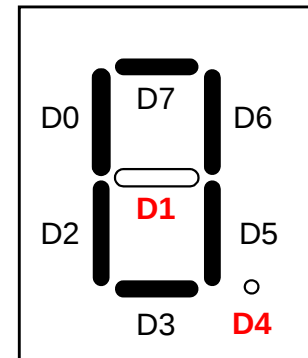
Programa principal

alternativa 1

```
#define SEGS (*(volatile unsigned char *)...) ..... declaración del nemotécnico para
la dirección de escritura en el display

void main( void )
{
    unsigned char i;
    unsigned int j; } declara variables índice

    SEGS = 0xff; ..... apaga todos los segmentos escribiendo 0xff en la dirección en donde se mapea el display
    while( 1 )
    {
        for( i=0; i<16; i++ ) {
            for( j=0; j<300000; j++ ); ..... pequeño retardo
            switch( i ){
                case 0x0:
                    SEGS = 0x12; break; // 0b00010010 ..... visualiza el 0
                case 0x1:
                    SEGS = ...; break;
                ...
                case 0xf:
                    SEGS = ...; break;
                default:
                    SEGS = ...; break;
            }
        }
    }
```



indefinidamente visualiza
los números del 0x0 al 0xF

Programa principal

alternativa 2



- Una solución más compacta sería:

```
#define SEGS (*(volatile unsigned char *)...)

const unsigned char hex2segs[16] = {0x12, ...}

void main( void )
{
    unsigned char i;
    unsigned int j;

    SEGS = 0xff;
    while( 1 )
        for( i=0; i<16; i++ ) {
            for( j=0; j<300000; j++ );
            SEGS = hex2segs[i];
        }
}
```

contiene los patrones de segmentos de cada dígito hexadecimal

- En cada caso, el programa ocupa:
 - o 464 B: primera alternativa con switch
 - o 136+16 B: segunda alternativa con array



Acerca de *Creative Commons*



■ Licencia CC (*Creative Commons*)

o Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:



Reconocimiento (*Attribution*):

En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.



No comercial (*Non commercial*):

La explotación de la obra queda limitada a usos no comerciales.



Compartir igual (*Share alike*):

La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Más información: <https://creativecommons.org/licenses/by-nc-sa/4.0/>