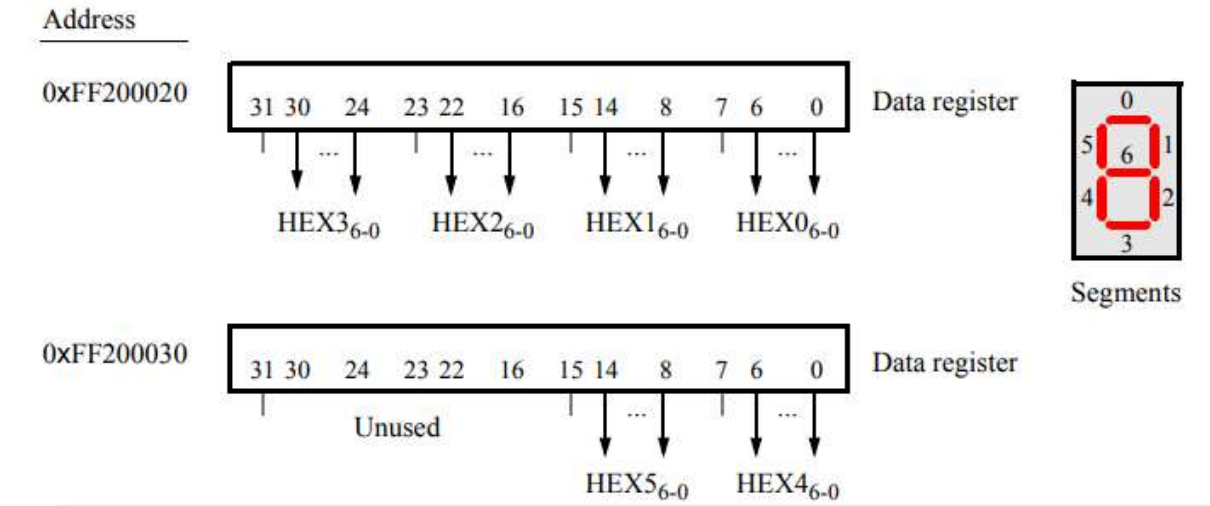
ARM - Display 7 segmentos

Simulador: https://cpulator.01xz.net/?sys=arm-de1soc

- Un display de 7 segmentos se puede utilizar para mostrar números y algunas letras.
- Por lo tanto, la salida es más un solo LED. Son 7 leds.
- El principio para programar el display 7 segmentos es el mismo que para el LED.

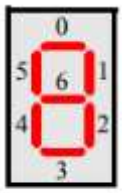


- Observamos que hay 6 display 7 segmentos y que cada uno tiene asignado 7 bits, uno para cada segmento.
- Cada uno de los 6 conjuntos de 7 bits está separado por un bit

del siguiente conjunto.

- La dirección de los display 7 segmentos es 0xFF200020
- Los 7 segmentos de cada display están controlados cada uno por 1 bit en el dato almacenado en la memoria
- La asignación de las posiciones de bits a los segmentos se muestra en la imagen de la derecha. Por ejemplo, el bit 0 controla la barra superior en el display 1.
- Los 7 bits menos significativos controlan los segmentos del display 0, los siguientes 7 bits se encargan del display 1 y así sucesivamente.

Ej.1 a. ¿Qué números van en cero (apagado) para un 2?
b. ¿Cuál es el patrón de bits que permite mostrar un 2 en un display 7 segmentos?



Segments

a)
El 5 y el 2

b)
1011011

Ej. 2 Completar la siguiente tabla:

NÚMERO	MÁSCARA
0	
1	
2	
3	
4	1100110
5	
6	
7	

8	
9	

@Escribir el número 2 en un display 7 segmentos

```
.global _start
_start:
        LDR R0, =0xFF200020        // Cargar la dirección de memoria 0xFF200020 en R0
        MOV R1, #0x5B              // Cargar el valor 0x5B (1011011 en binario) en R1
        STR R1, [R0]               // Almacenar el valor de R1 en la dirección R0

    // Fin del programa
    SWI 0x11                       // Llamada al sistema para salir
```

El siguiente código muestra el número 2024 en los 4 displays 7 segmentos de más a la derecha.

```
.global _start
_start: LDR     R0, =HEXBASE    @ physical address of the HEX
        MOV     R2, #0b1011011  @ bitmask for pattern of a '2'
        MOV     R3, R2, LSL #24 @ display: 2
        MOV     R2, #0x3f       @ bitmask using hex-repres.
        ORR     R3, R2, LSL #16 @ display: 0
        MOV     R2, #91         @ bitmask using a decimal no.
        ORR     R3, R2, LSL #8  @ display: 2
        MOV     R2, #6          @ again usin a decimal value
        ORR     R3, R2, LSL #0  @ display: 1
        STR     R3, [R0]
.data
.equ    HEXBASE,        0xFF200020
```

```
.data
.equ DISPLAYS, 0xFF200020
.global _start

-start:  LDR    R0, =DISPLAYS                @ dirección de memoria de los displays
                MOV    R2, #0b1011011        @ máscara para un 2
                MOV R3, R2, LSL #24          @ display 2
                MOV R2, #0x   3F             @ máscara en hexadecimal
```

```
ORR R3, R2, LSL #16     @ display 0
MOV R2, #91             @ máscara en decimal
ORR R3, R2, LSL #8      @ display 2
MOV R2, #6              @ máscara en decimal
ORR R3, R2, LSL #0      @ display 1
STR R3, [R0]            @ máscara en decimal
```