



Práctica 2

- Máquinas de estados finitas



Práctica 2 avanzada

- Ampliar la parte básica de la máquina de estados de la práctica 2 para que se lleve la cuenta de cuántas veces hubo una respuesta “rápida”, “intermedia” o “lenta”.
 - Se instanciarán 3 contadores, que irán conectados a 3 conversores 7 segmentos (cuyo código VHDL os doy en el Campus Virtual).
 - Los 3 conversores a 7 segmentos irán conectados a 3 nuevas salidas (NumR, numM, numL).

```
numR: out std_logic_vector (6 downto 0);
```

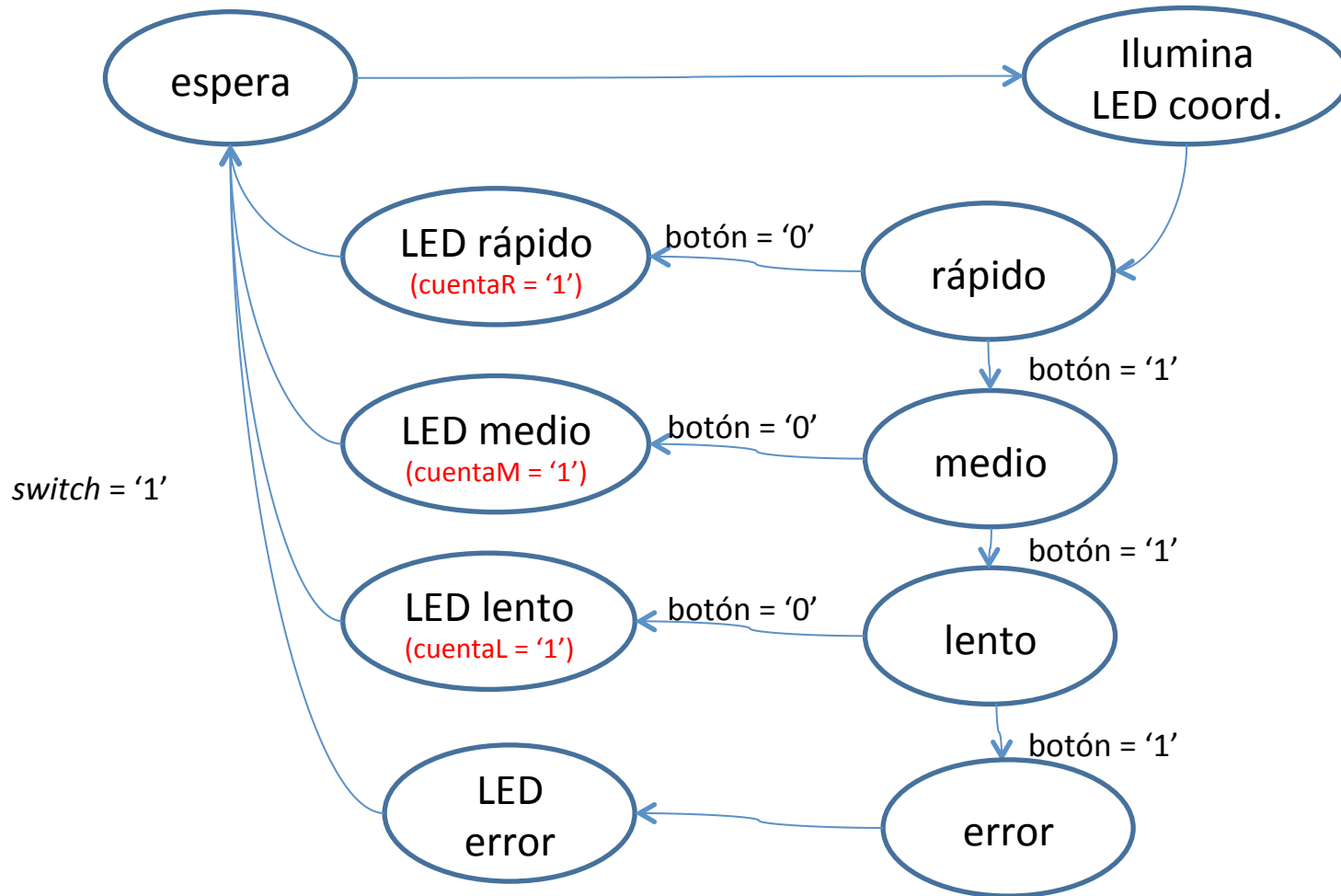
```
numM: out std_logic_vector (6 downto 0);
```

```
numL: out std_logic_vector (6 downto 0);
```

- Estos contadores serán controlados por la máquina de estados.
- Seguiremos utilizando el divisor de frecuencias.



Nuevo diagrama de estados

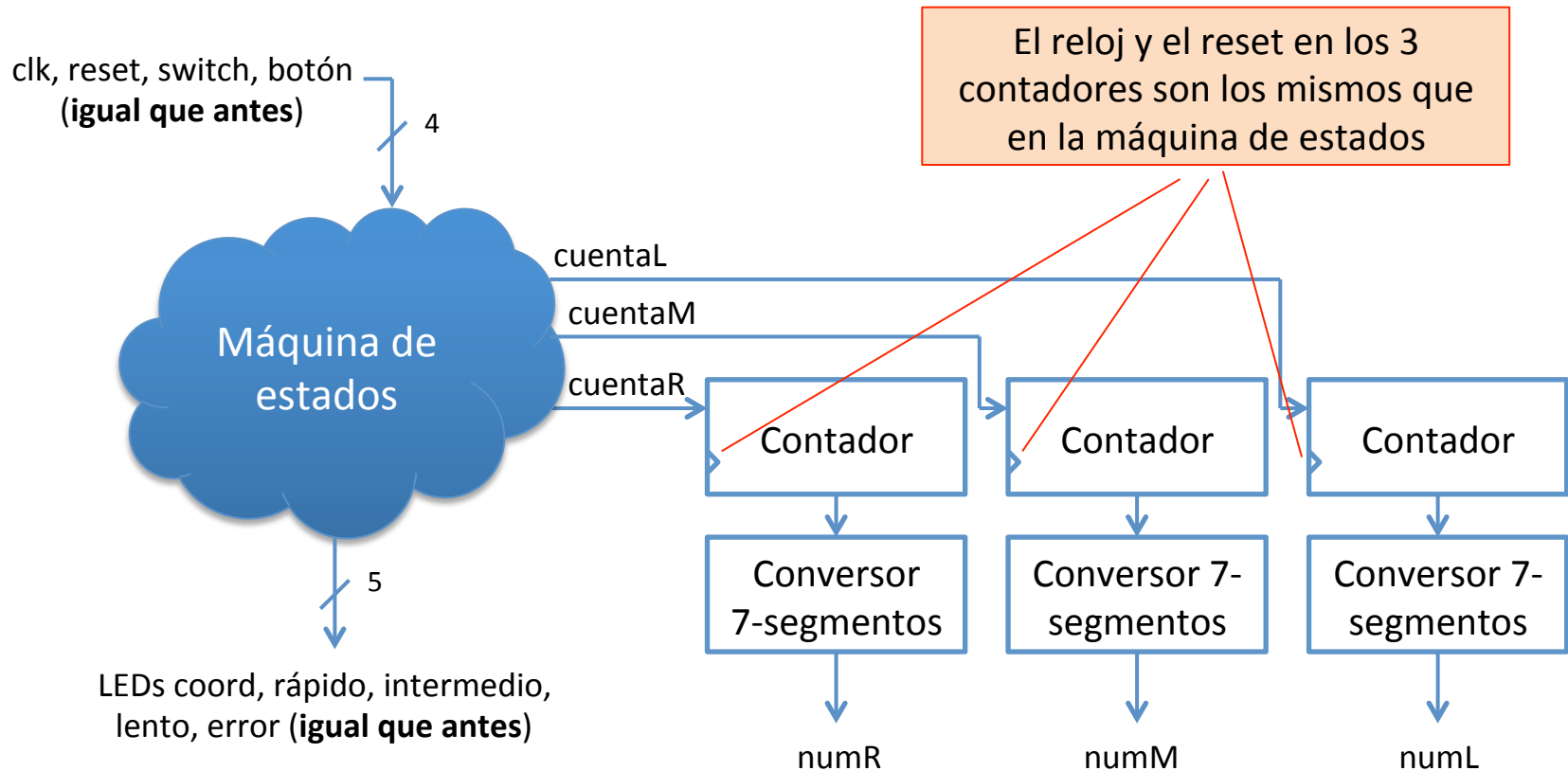


En los estados en los que no se especifica nada, "cuentaR", "cuentaM" y "cuentaL" **deben** estar a '0'



Práctica 2 avanzada

- El diagrama RTL de la nueva versión de la práctica sería el siguiente:





Práctica 2 avanzada

- En el fichero .ucf tendréis que asignar los pines de los 3 displays disponibles (2 en la placa extendida y 1 en la placa superior). Por ejemplo:

#Display 7-segmentos de la parte izquierda de la placa extendida

```
NET numR<0> LOC=H14;  
NET numR<1> LOC=M4;  
NET numR<2> LOC=P1;  
NET numR<3> LOC=N3;  
NET numR<4> LOC=M15;  
NET numR<5> LOC=H13;  
NET numR<6> LOC=G16;
```



Práctica 2 avanzada

- En cada uno de los 10 posibles dígitos (de 0 a 9), la salida “numXXX” tendrá un valor diferente. Por ejemplo, para mostrar un 0, sería así:
 - numR <= “0111111”;
- Asumiendo que la distribución de los 7 segmentos del display es así (viene comentada en el fichero .ucf):

```
#
#           S0
#           ---
#      S5   |           |s1
#           S6
#           ---
#      S4   |           |S2
#           S3
```

- La conversión entre los números y los valores del display 7-segmentos ya está realizada en el conversor a 7-segmentos, así que no hay que preocuparse por eso en esta práctica