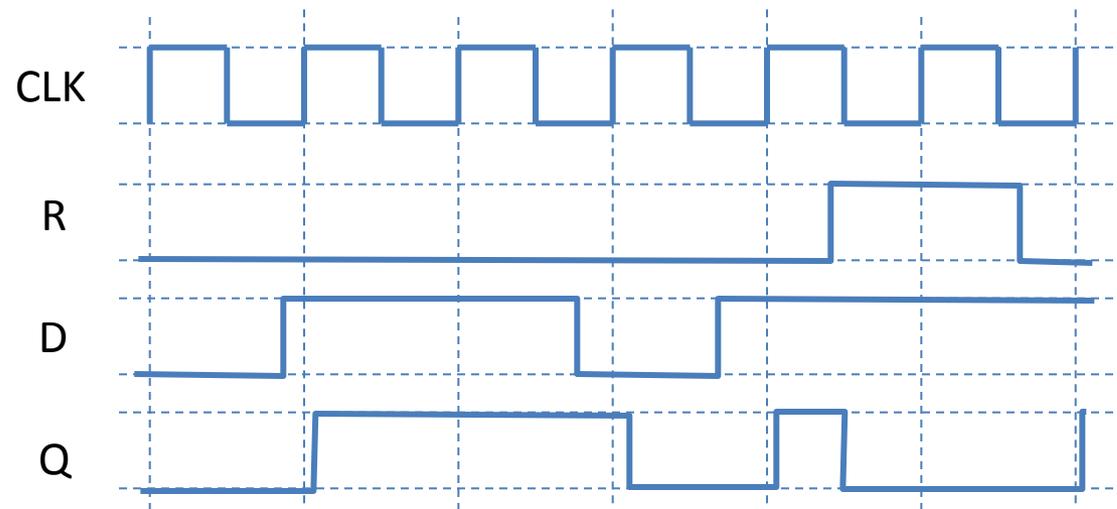
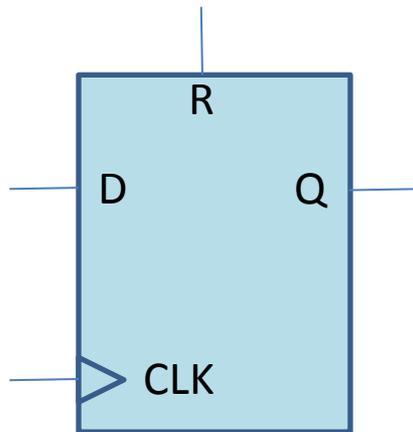


Diseño electrónico digital integrado: *Back to basics*

Flip-flop D con Reset Asíncrono

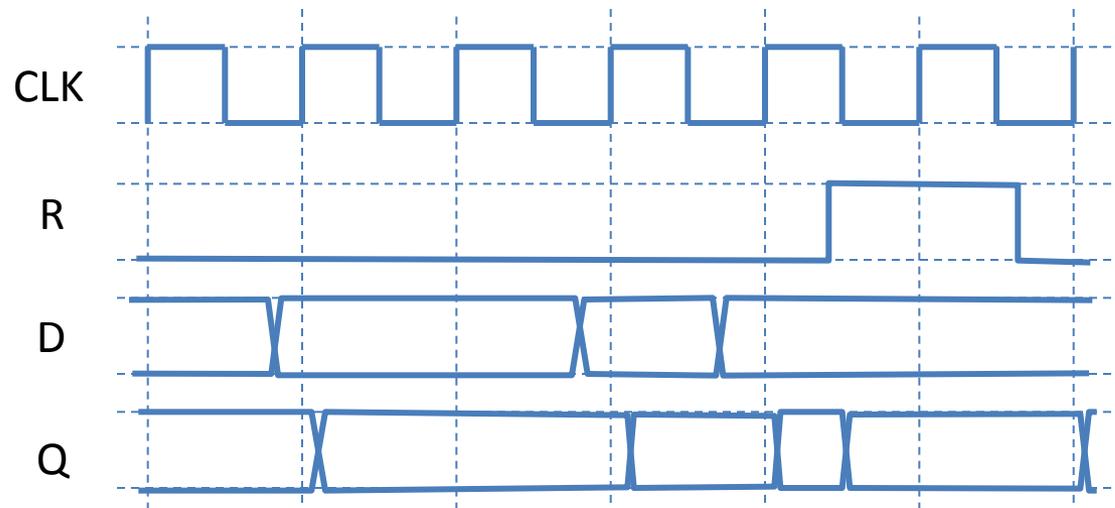
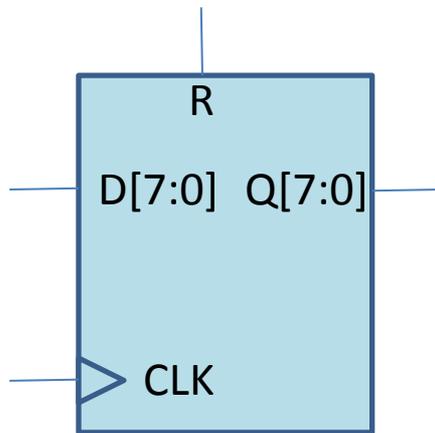
- Captura un dato con el flanco de reloj
- La pata de reset lo vuelve a su valor inicial inmediatamente.



El clock puede ser activo en flanco de subida (lo normal) o de bajada (rarísimo).
El reset puede estar presente o no, y ser activo a nivel alto o a nivel bajo.
El reset puede poner el flip-flop a 0 o a 1.

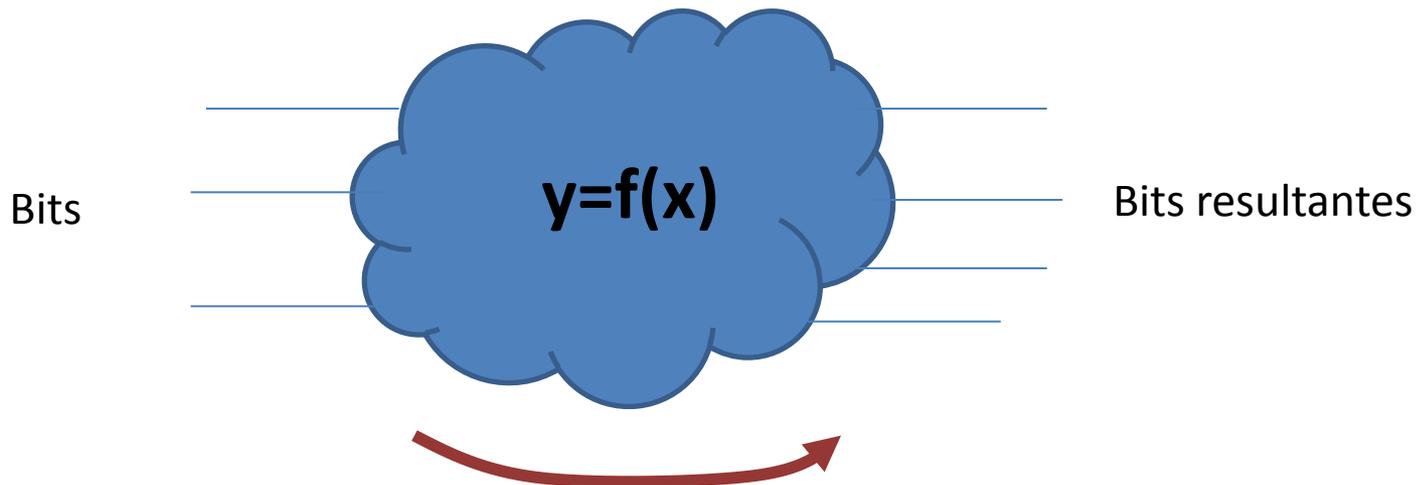
Registro (Register)

- Conjunto de 1 o más flip-flops en paralelo, se comportan igual pero pueden tener valores de más de un bit



Lógica combinacional (combinational logic)

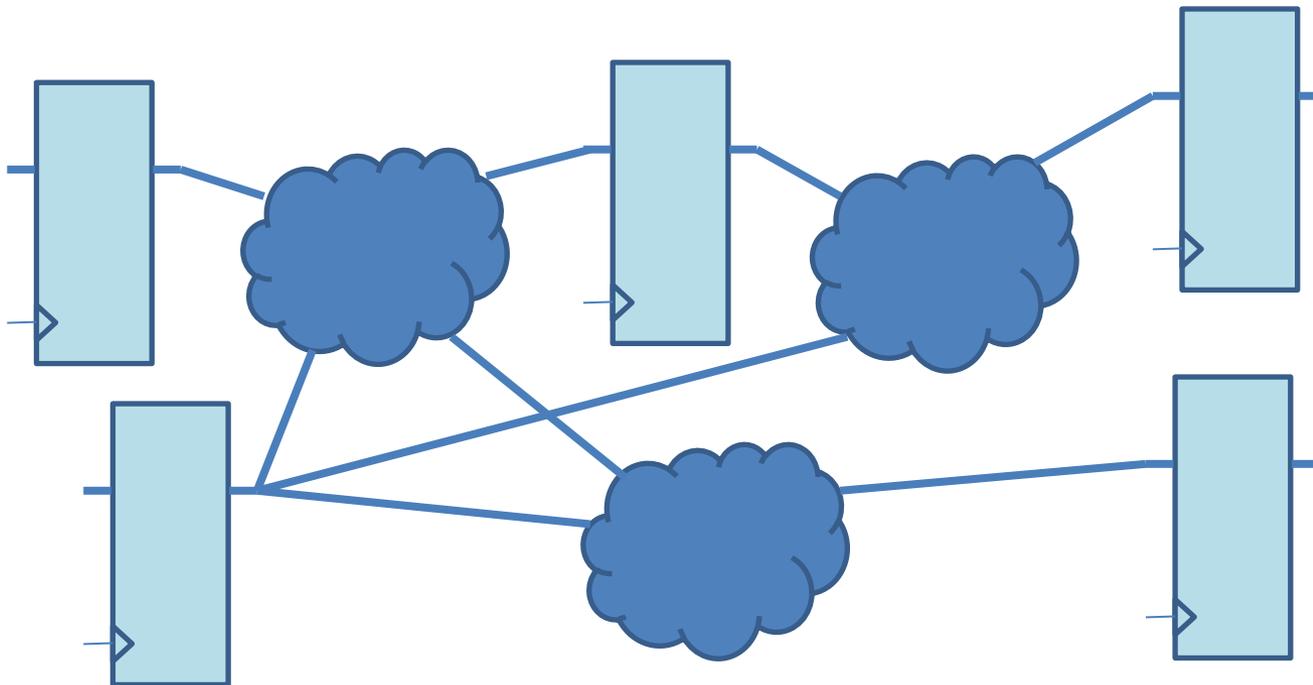
- Porción de un circuito que aplica una función lógica a un conjunto de bits de entrada para dar un conjunto de bits de salida, por propagación directa (sin esperar ningún reloj)



Retardo de propagación
Depende de lo rápida que sea la tecnología
(transistores, pistas...), NO de un reloj.

RTL (Register Transfer Level)

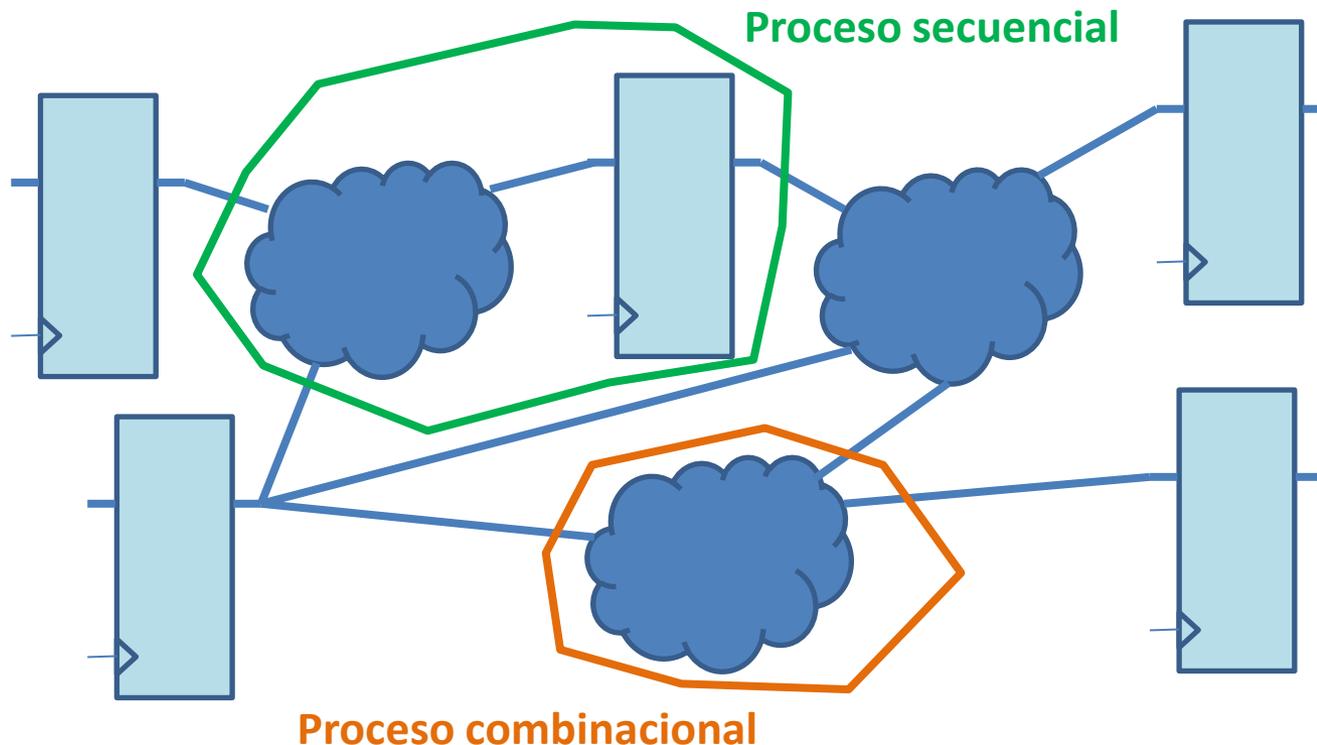
- Tipo de descripción “behavioral” de un circuito que se basa exclusivamente en transferir datos entre registros bien definidos (tipo “D”). Como casos particulares, puede existir solamente lógica combinacional o solamente registros



¡Único tipo de descripción *synthesizable*!

Procesos secuenciales y combinacionales

- En descripciones HDL, decimos que un proceso es “secuencial” cuando engloba registros y opcionalmente lógica combinacional a sus entradas, mientras que un proceso “combinacional” sólo tiene lógica combinacional.



Flip-flop (FF) D con Reset Asíncrono

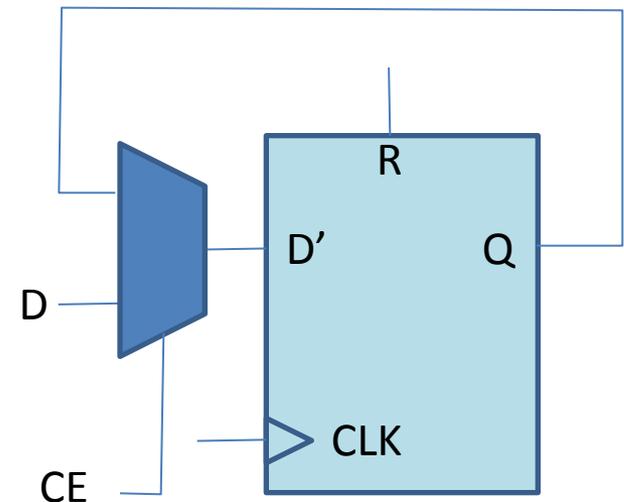
- VHDL:

```
process (clk, rst)
begin
    if rst='1' then
        q <= '0';
    elsif clk'event and clk='1' then
        q <= d;
    end if;
end process;
```

Flip-flop D con Reset Asíncrono

- Con clock enable:

```
process (clk,rst)
begin
  if rst='1' then
    q <= '0';
  elsif clk'event and clk='1' then
    if ce='1' then
      q <= d;
    end if;
  end if;
end process;
```



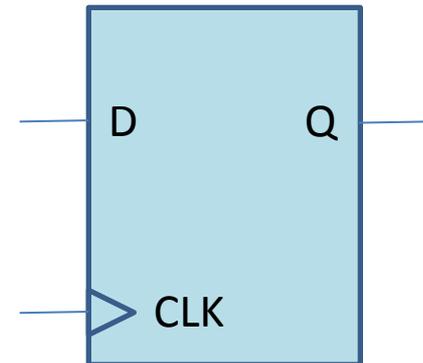
¡No es más que lógica combinacional añadida!

No despistarse: recordar el elemento básico,
el FF D, con o sin reset asíncrono.

Flip-flop D sin Reset Asíncrono

- VHDL:

```
process (clk)
begin
    if clk'event and clk='1' then
        q <= d;
    end if;
end process;
```



En diseño ASIC se evita lo máximo posible el usar FF sin reset.

En FPGA su uso es más frecuente.

→ Si se quiere hacer diseño portable de FPGA a ASIC, lo mejor es usarlo siempre.

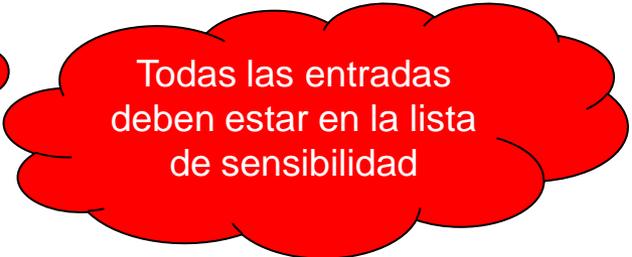
Logica combinacional

- VHDL: *process* que en su *lista de sensibilidad* tiene las entradas de la función combinacional

```
architecture uam of mux is
begin
  process(a,b,sel)
  begin
    if sel='1' then
      y <= a;
    else
      y <= b;
    end if;
  end process;
end uam;
```



Se debe asignar siempre (en todos los casos) a la salida un valor



Todas las entradas deben estar en la lista de sensibilidad

Procesos VHDL

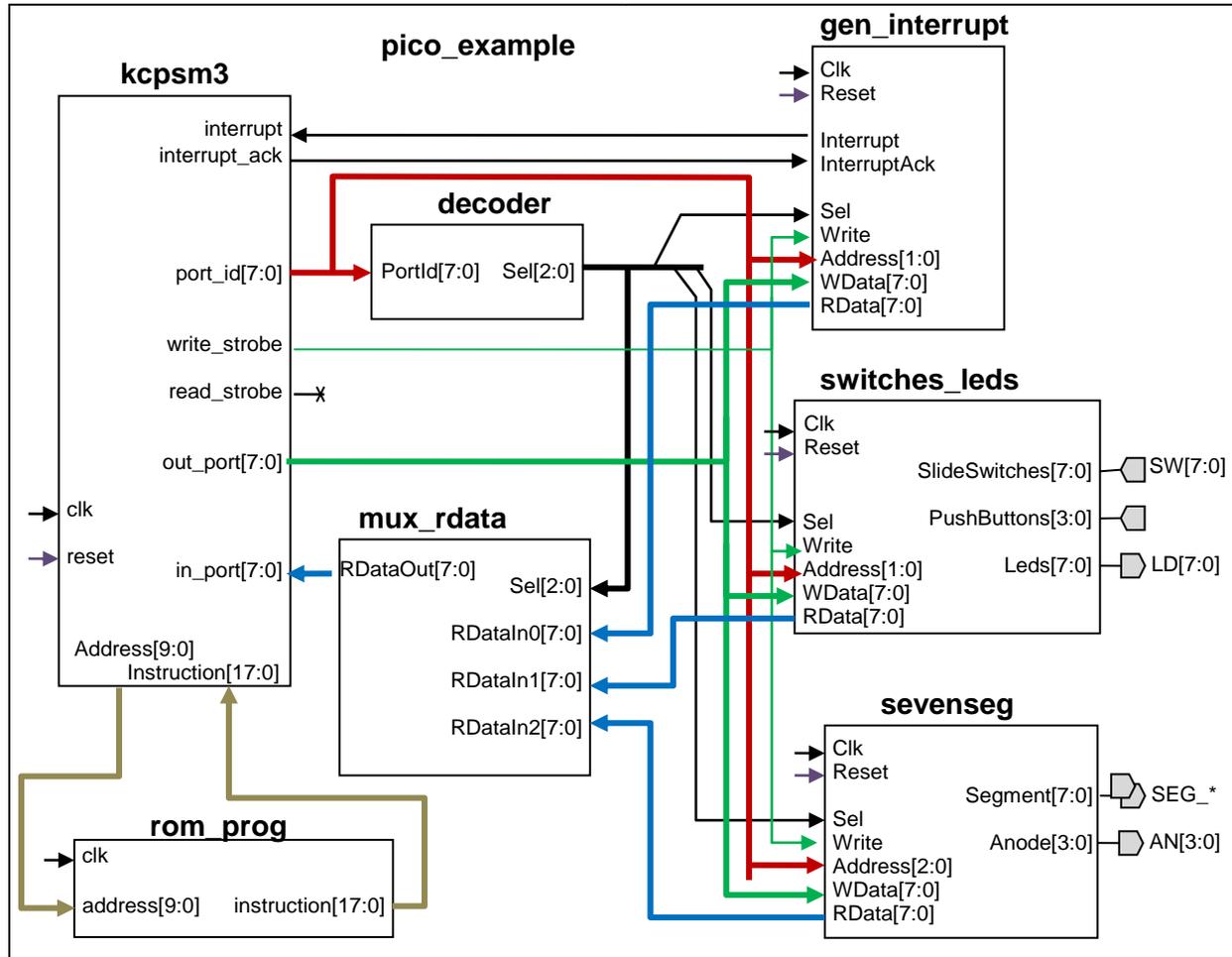
- En una descripción RTL VHDL todo son procesos combinacionales o secuenciales (FF), ejecutándose concurrentemente.
- Las asignaciones concurrentes nos permiten escribir más rápidamente procesos combinacionales.

```
s <= (a and b) + c;  
  
with a+b select  
  s <= d when "0000",  
  e when "1010",  
  '0' when others;
```

- Todo esto se hace con *signals*.
- Las *variables* VHDL permiten tener variables temporales dentro de los *process* (al estilo de la programación software).

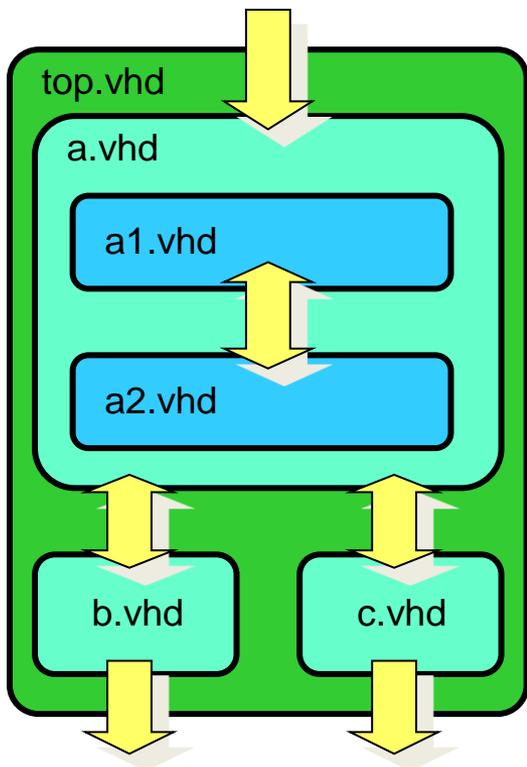
Descripción estructural

- Ejemplo: top-level práctica laboratorio *PicoBlaze*:



Descripción estructural

- VHDL: componentes e instancias



```
ENTITY top IS PORT  
( ... )  
END top;
```

```
ARCHITECTURE structural OF top IS  
signal s1,s2 : std_logic;
```

```
COMPONENT a  
PORT ( entrada : IN std_logic;  
       salida  : OUT std_logic );  
END COMPONENT;
```

```
begin
```

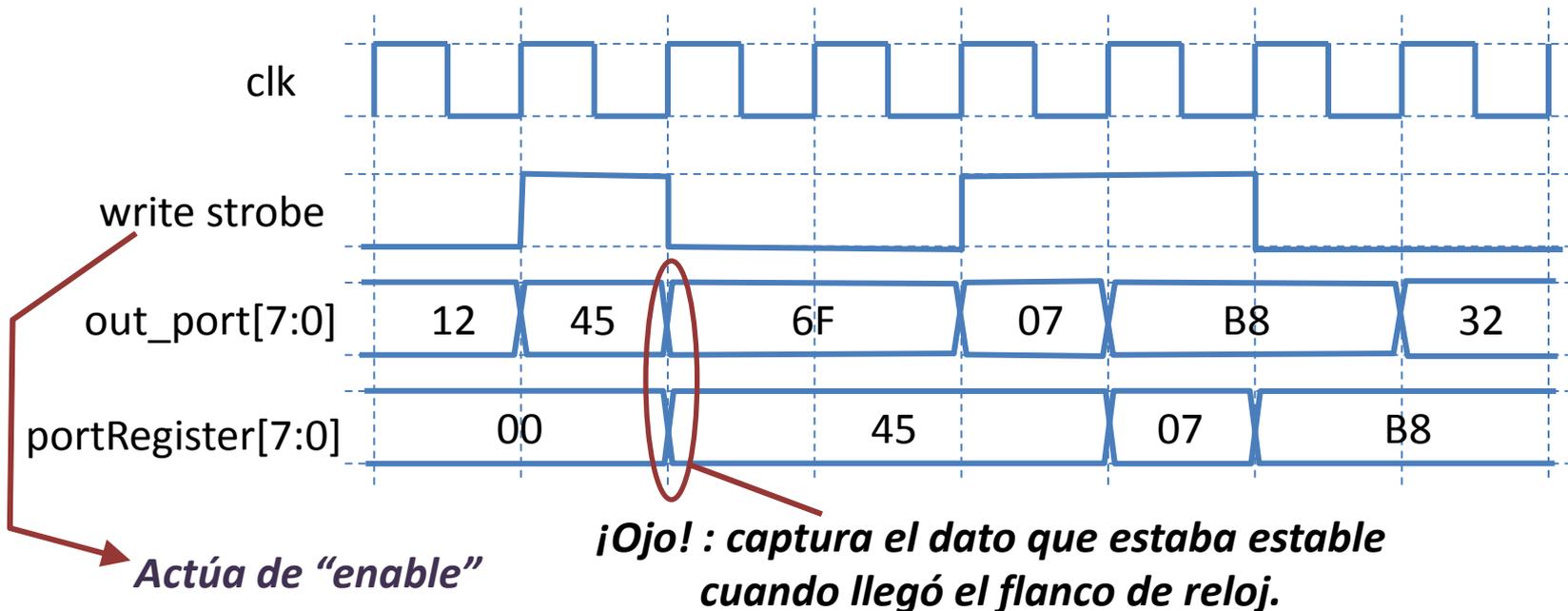
```
u1: a PORT MAP  
      (entrada=>s1, salida=>s2);
```

```
....
```

```
end jerarquica;
```

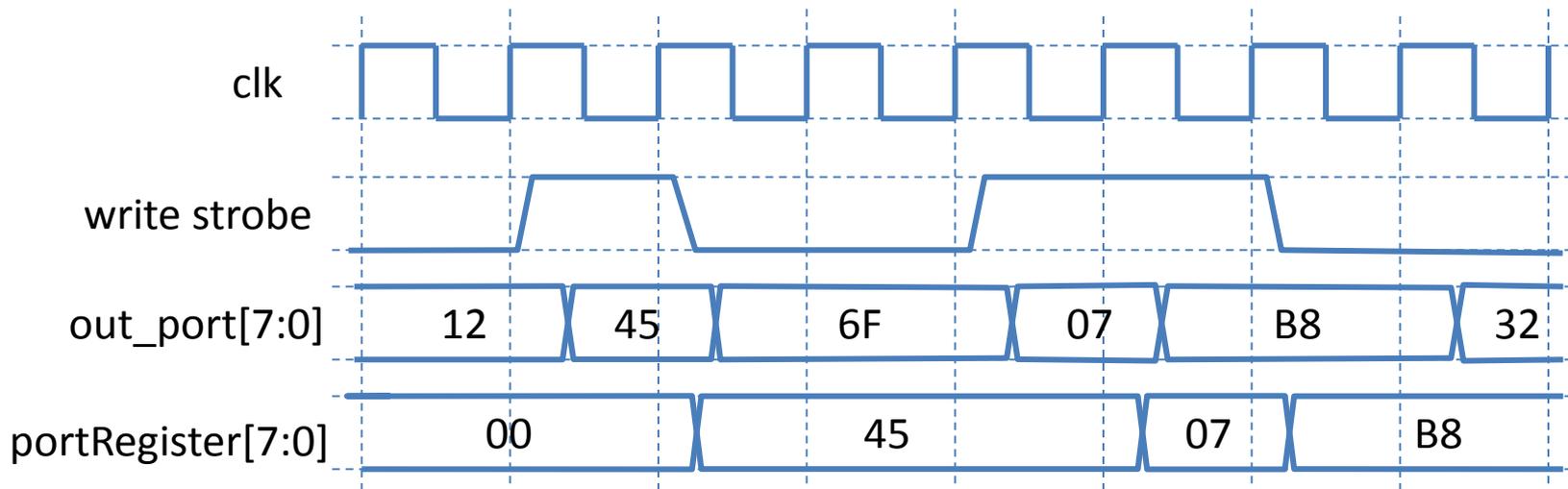
Cronogramas

- Ejemplo: escritura del micro en puerto
 - El dato va de un registro del micro a un registro de un periférico.



Cronogramas

- Ondas más realistas:



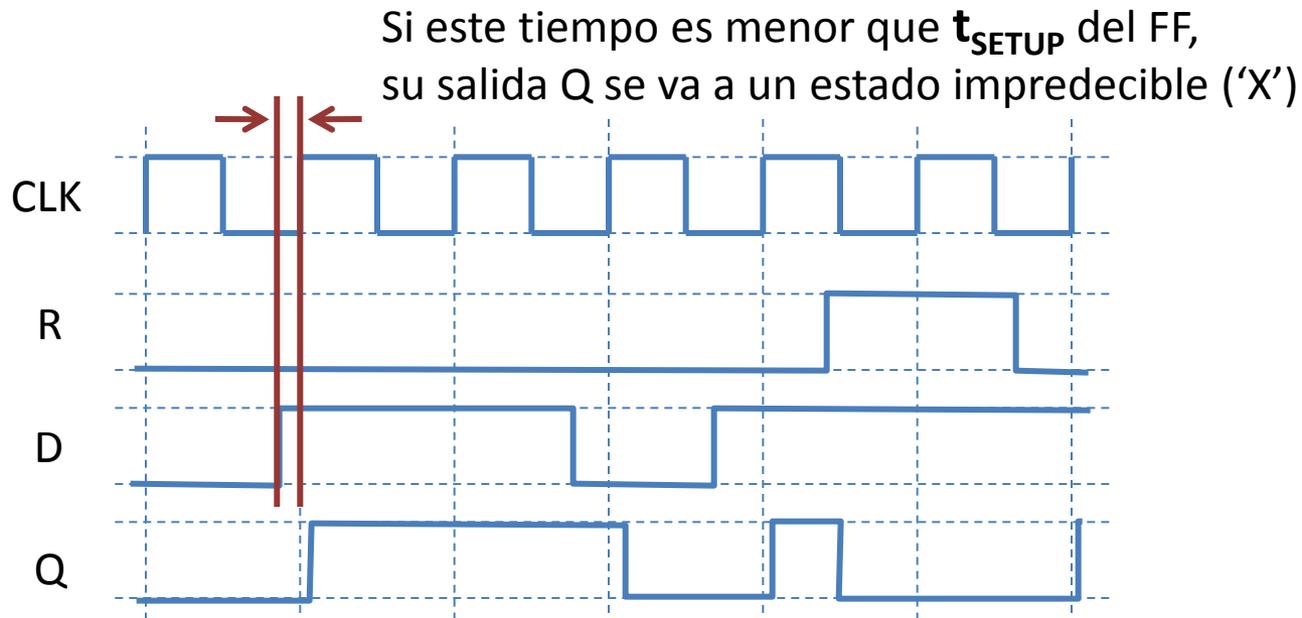
Así es más fácil tener claro qué valor carga el registro, pero...:

- En simulación sin tiempos nos vamos a encontrar la otra representación
- En el cuaderno es más cómodo usar la otra representación 😊

↳ *Importante: recordar que el papel y el lápiz también existen, y el papel cuadriculado es una herramienta de diseño a tener en cuenta...*

Setup

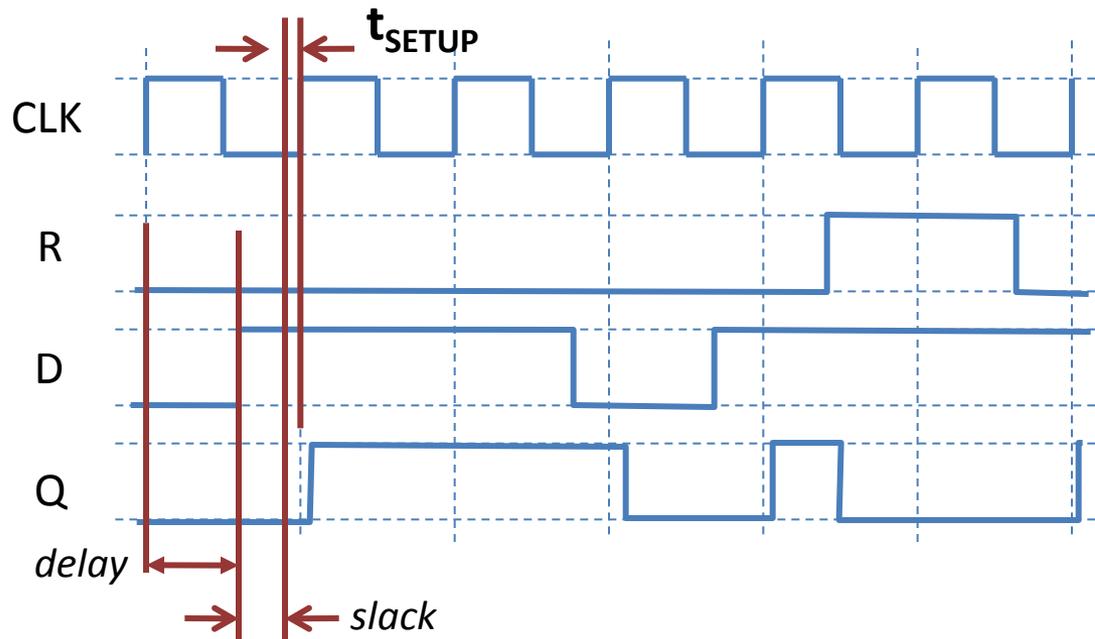
- Los FF no son “perfectos”, necesitan que el dato de entrada esté estable un mínimo de tiempo



Hold : Mismo concepto, pero el requerimiento es de estabilidad de *D* tras el flanco de reloj.

Slack y frecuencia máxima

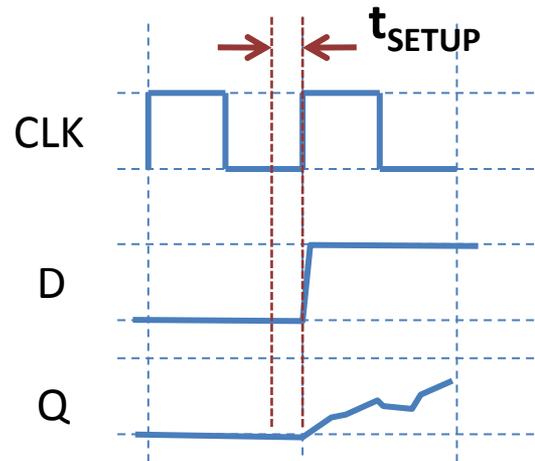
- El slack es el tiempo que nos sobra para poder cumplir con los requisitos (si es negativo, mal)



Si encogemos más y más el periodo de reloj, el retardo (*delay*) va a ser el mismo y el tiempo de *setup* también, pero el *slack* disminuye. Cuando es 0 hemos alcanzado la frecuencia máxima para ese “*path*”.

Metaestabilidad

- Puede aparecer al tomar como dato una entrada externa al sistema o proveniente de otro “dominio de reloj”.
- El cambio del dato se produce a la vez que el de reloj, y la salida se comporta de forma anómala. Puede provocar fallos graves en los circuitos.



- Se soluciona poniendo una cadena de FFs

