Recomendaciones para diseño digital

Diseño Lógico II

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería
Universidad de la República

2012

Project Reliability Guidelines Asynchronous Inputs

- respetar setup y hold
- problemas de metaestabilidad
- señal asíncrona debe entrar a 1 solo FF

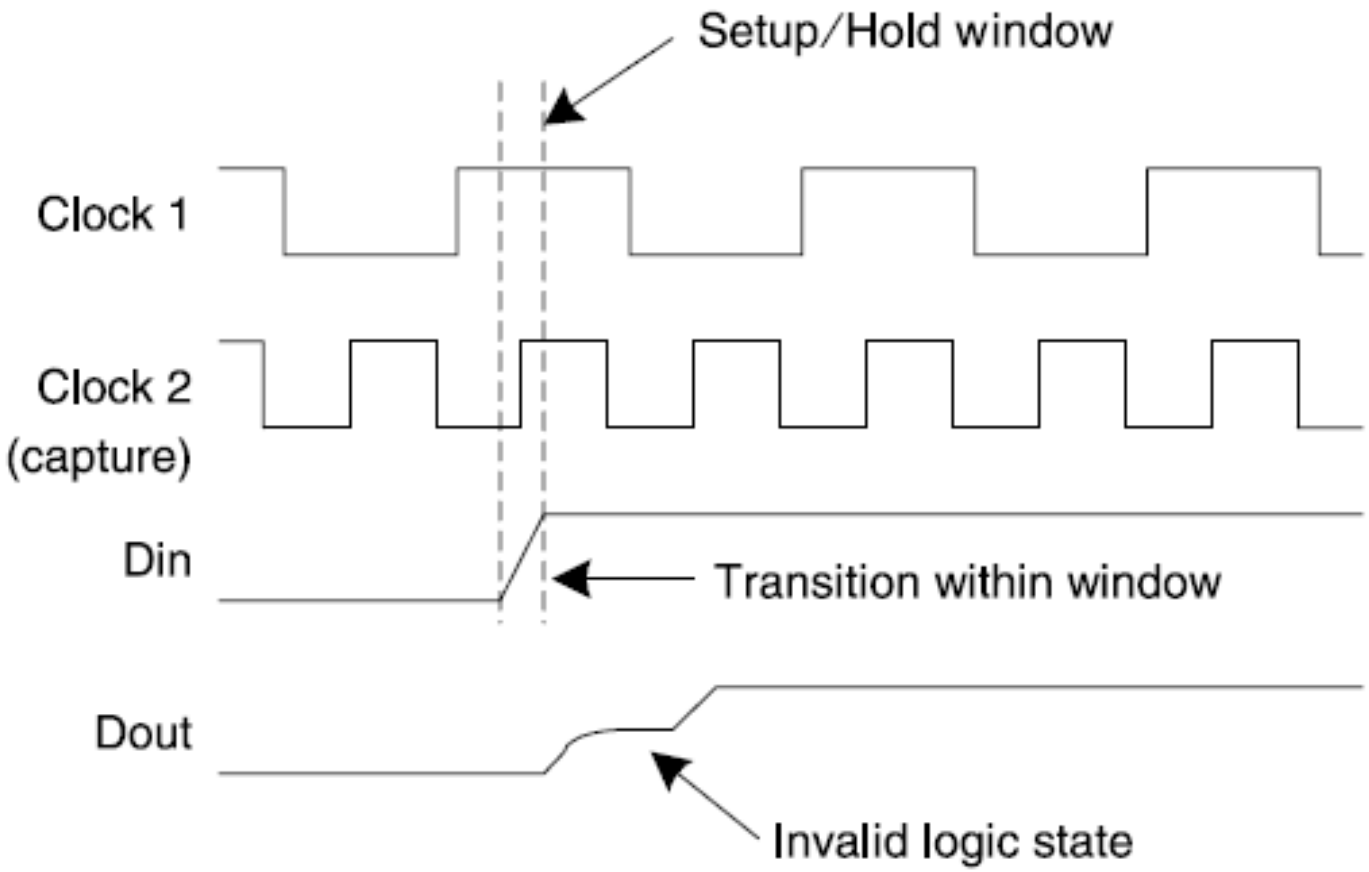
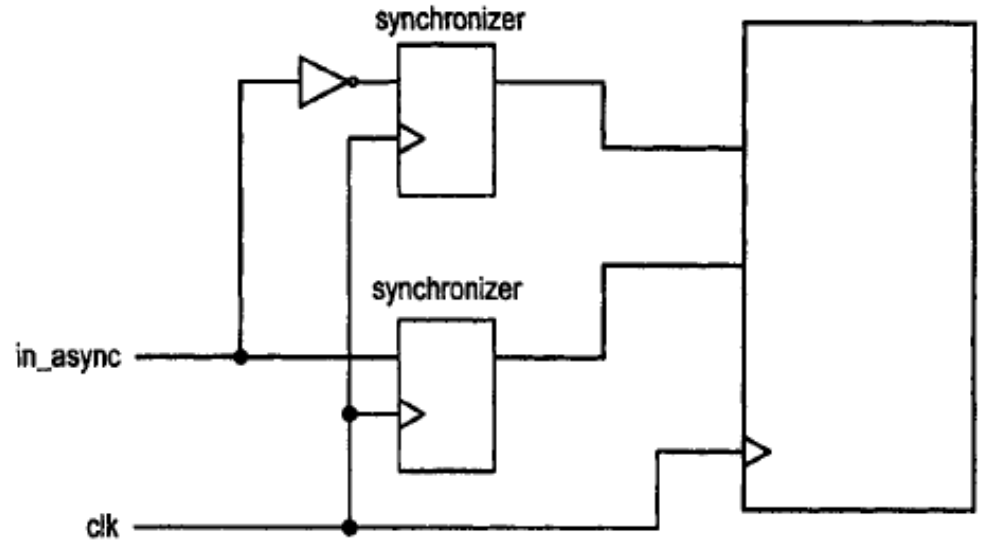
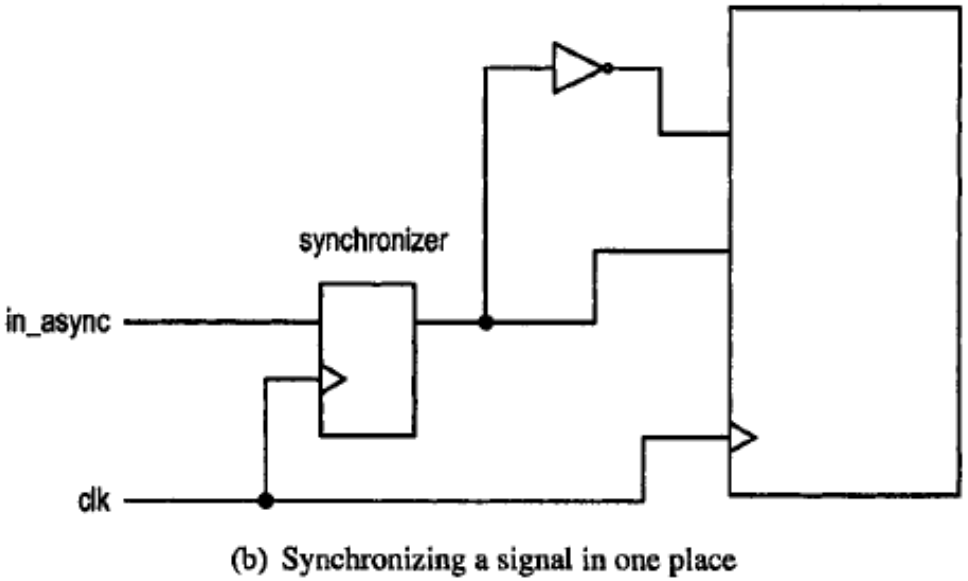


Figure 6.6 Metastability caused by timing violation.

(del libro “Advanced FPGA Design Architecture, Implementation, and Optimization”, Steve Kilts)



(a) Synchronizing a signal in two places



(b) Synchronizing a signal in one place

Figure 16.10 Synchronization at multiple places.

- MAL
- Diferentes FF pueden capturar diferentes valores.

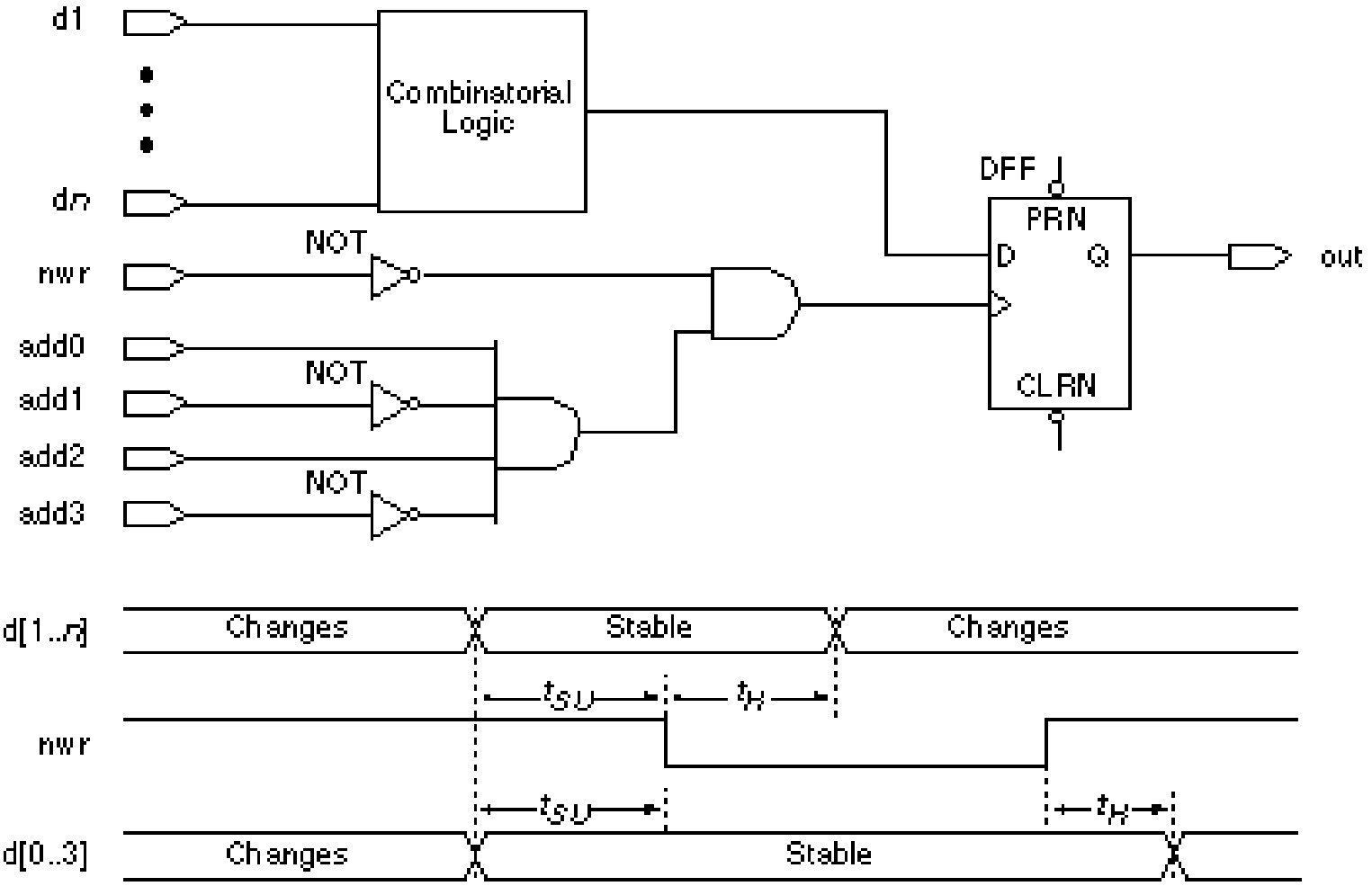
- OK

Figura de *"RTL Hardware Design Using VHDL"*, Chu

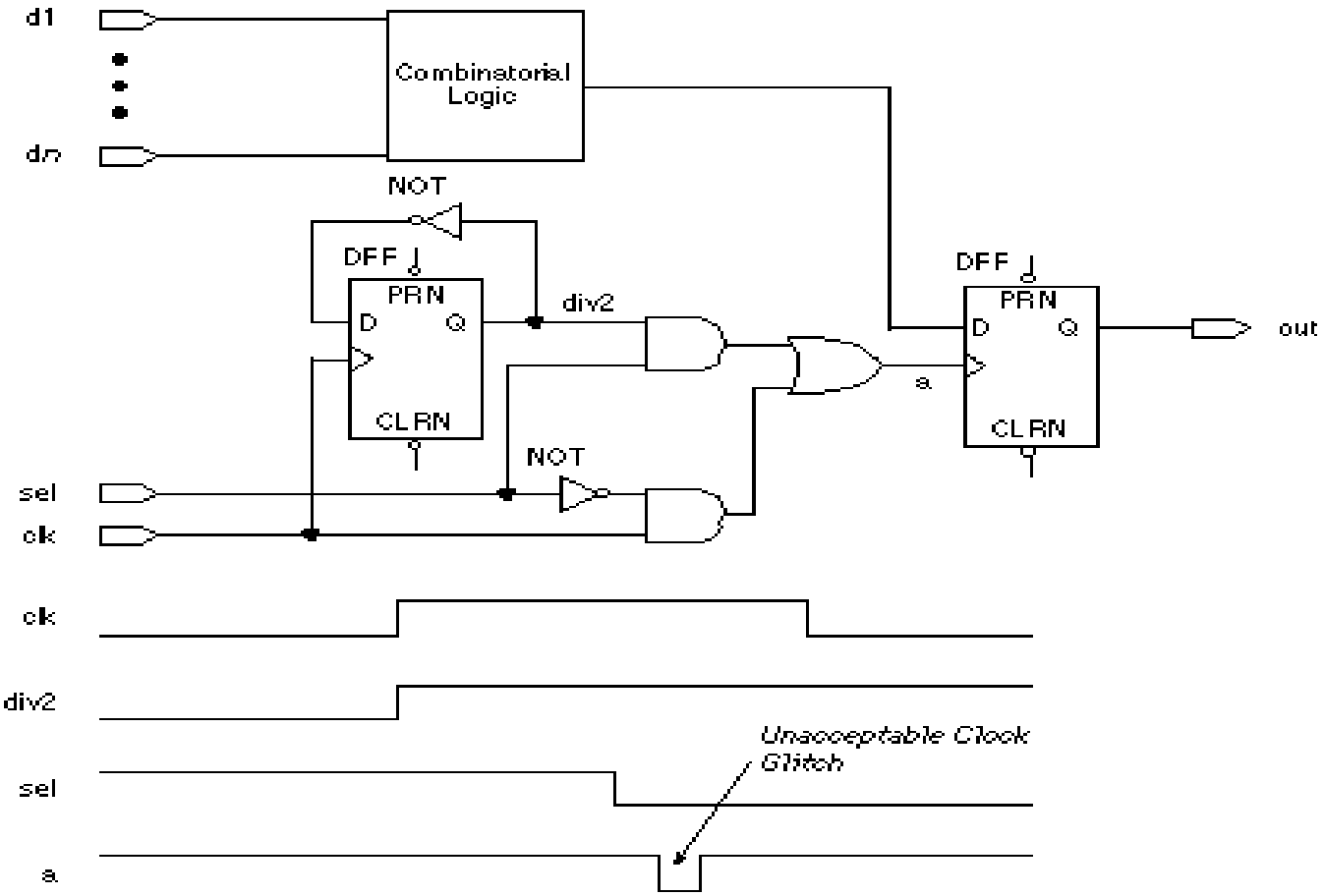
Project Reliability Guidelines
Clock Configurations

- Global Clocks. Primitivas GLOBAL y altclklock
- Gated Clocks
  - Reloj generado con una sola compuerta
  - Mucho cuidado con glitches
  - Gated Clock Examples *
- Multi-Level Clocks
  - Reloj generado con lógica en varios niveles.
  - Multi-Level Clock Example *
  - Des-recomendado por glitches.
  - Alternativa: flip-flops con entrada ENABLE

Gated clock


d1
dn
Combinatorial Logic
DFF
PRN
D
Q
CLRN
out
nwr
NOT
add0
add1
add2
add3
NOT
NOT
d[1..n]
Changes
Stable
Changes
tSU
tH
nwr
tSU
tH
add[0..3]
Changes
Stable

Multilevel clock



Sel clk \ div2	0	1
00		
01		1
11	1	1
10	1	

Señales de reloj

- Utilizar pads y lineas de distribución adecuadas para asegurar un bajo ***skew***.
- Mantener la generación de relojes y señales de control en bloques separados.
- Utilizar la menor cantidad de relojes diferentes. En caso de tenerlos, utilizarlos en un bloque para pasar de un dominio de reloj a otro (dual clock fifo).
- En chips de Altera, utilizar “Clock Control Block”
  - Salida a líneas globales.
  - Entrada desde pin, PLL o lógica interna en forma estática o dinámica.
  - Habilitación / Deshabilitación y selección

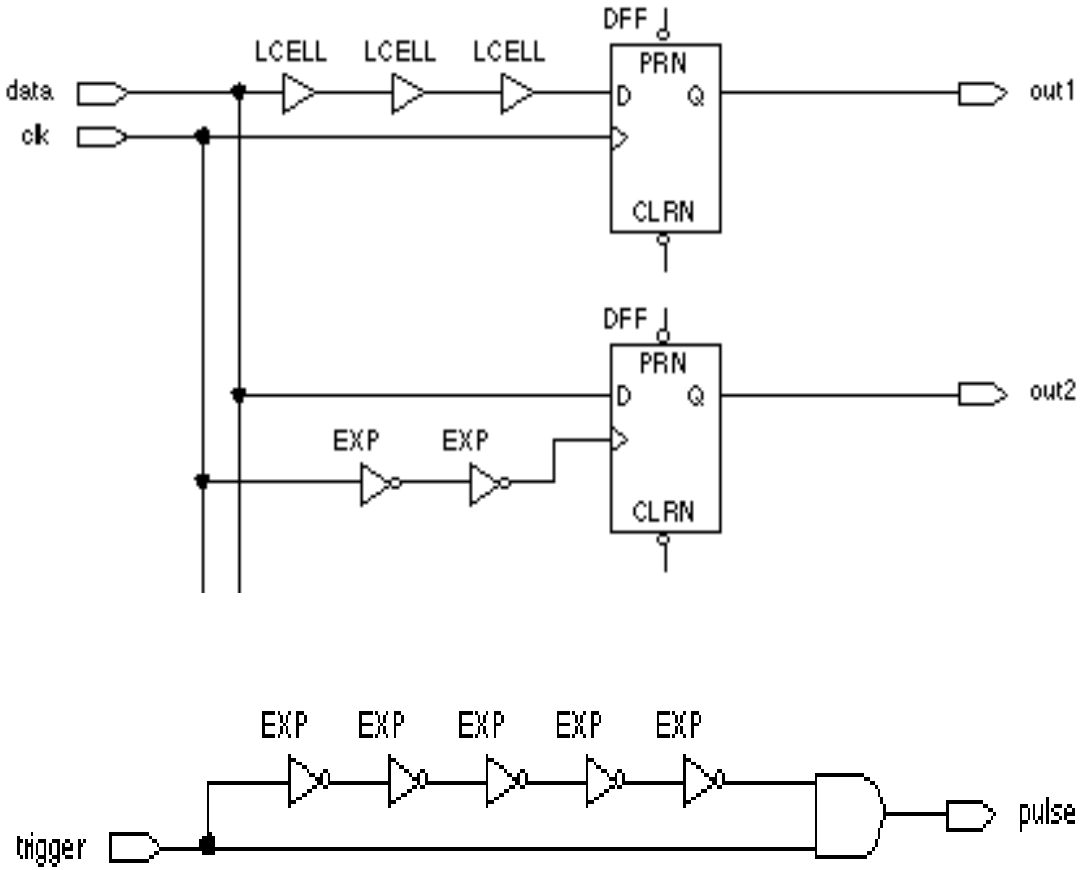
Never ever touch the clock!

(Unless you really know what you are doing.)

Project Reliability Guidelines
Delay Chains

- retardo que quiero introducir ex-profeso
- peligro de que el compilador lo elimine al minimizar
- No es una buena costumbre, no hay garantía de duración mínima de los retardos introducidos
- Delay Chain Examples
- Alternativa confiable: PLLs

Delay Chains



Project Reliability Guidelines
Latches

- En lo posible usar latches y FF del dispositivo
- Recomendación de Altera: Si es necesario armar un latch con compuertas, entonces usar las macrofunciones de Quartus (p. ej. nandltch y norltch)

Latches no deseados

- Omitir casos genera latch
  - "else" en if
  - "when others" en case
- std_logic
  - 9 valores posibles
- Asignar don't care
  - permite simplificar

```
PROCESS (a,b,c,sel)
 BEGIN
   if sel = "00" THEN
     oput <= a;
   ELSIF sel = "01" THEN
     oput <= b;
   -- …
   -- …
   ELSE
     oput <= "X"; --/
 END if;
END PROCESS;
```

Reset de sistema

- Asegurarse de que todos los FF del sistema se resetean.

- **Reset síncrono:**
  - (+): fácil de sintetizar, es una entrada más del sistema.
  - (-): requiere períodos “libres” de reloj luego de power-up
  - (-): consume más recursos

- **Reset asíncrono:**
  - (+): no requiere períodos libres de FF.
  - (+): No utiliza entrada de datos de los FF. No afecta timing
  - (-): requiere de caminos especiales (como clk)
  - (-): sensible a glitches

Reset de sistema

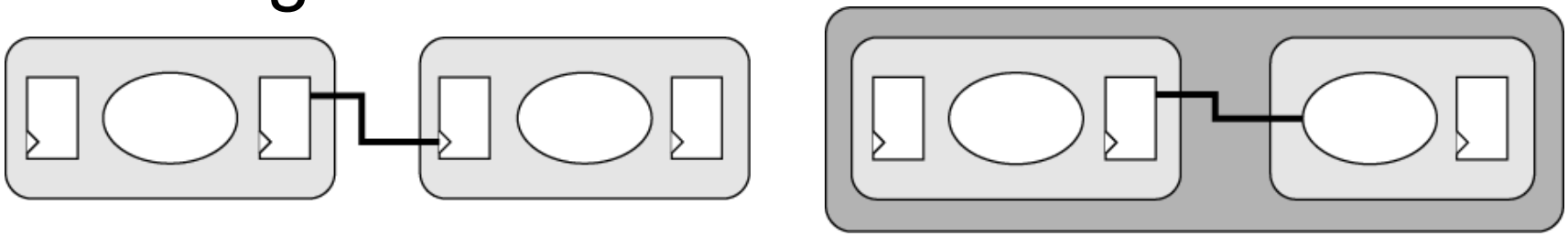
- SIEMPRE hay que tener un mecanismo para llevar el sistema a un estado conocido.
- Igual que con relojes
  - Usar señales GLOBAL para reset asíncrono
  - Evitar glitches y azares

Clear vs Reset

- Clear: borrar bloques de datos (síncrono)
- Reset: llevar control a estado conocido (asíncrono)

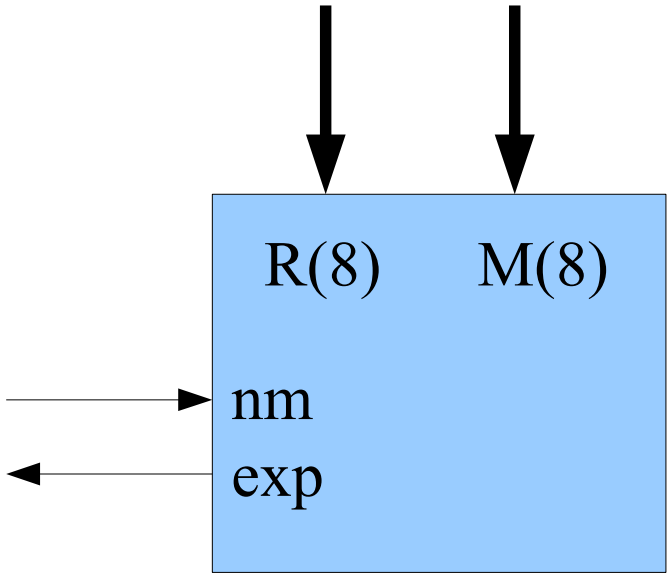
Interfaces registradas

- Evita problemas de tiempo entre bloques, acota tiempos de propagación.
- Acompañado de una especificación clara de interfaz mantiene acotado el diseño (facilita su análisis y detección de errores).
- Evita glitches en las salidas.



RTL

- Ejemplo: expendedor de refrescos



R: precio de refresco
M: valor de moneda insertada
nm: pulso que indica nueva moneda
exp: pulso para expender refresco

Descripción RTL

Inputs: R[8], M[8], nm

Outputs: exp

Memory: A[8]

0. A ← 0

1. A*nm ← A + M

   → (A==R, A!=R)/(2,1)

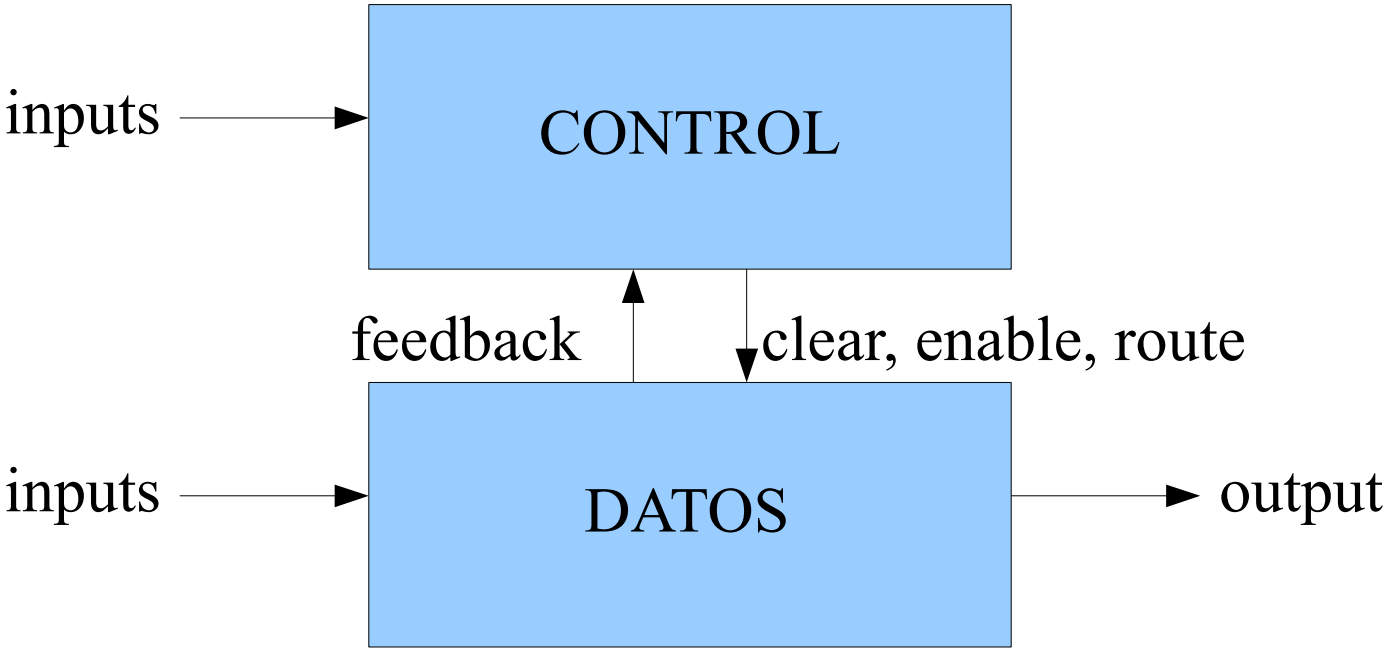
2. A ← 0

   exp = 1

ENDSEQUENCE

CONTROL RESET (0) END.

RTL


inputs
CONTROL
feedback
clear, enable, route
inputs
DATOS
output

Referencias

- Altera "*Project Reliability Guidelines, Max+PlusII help"*
- Altera "*AN 466: Cyclone III Design Guidelines*"
- Altera "Quartus II Handbook, Vol. 1: Design and Synthesis, Sect III: Design Guidelines"

- Chu, P.; *"RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability"*, *Wiley, John & Sons, Incorporated*, **2006**

- http://www.sm.luth.se/csee/courses/smd/098