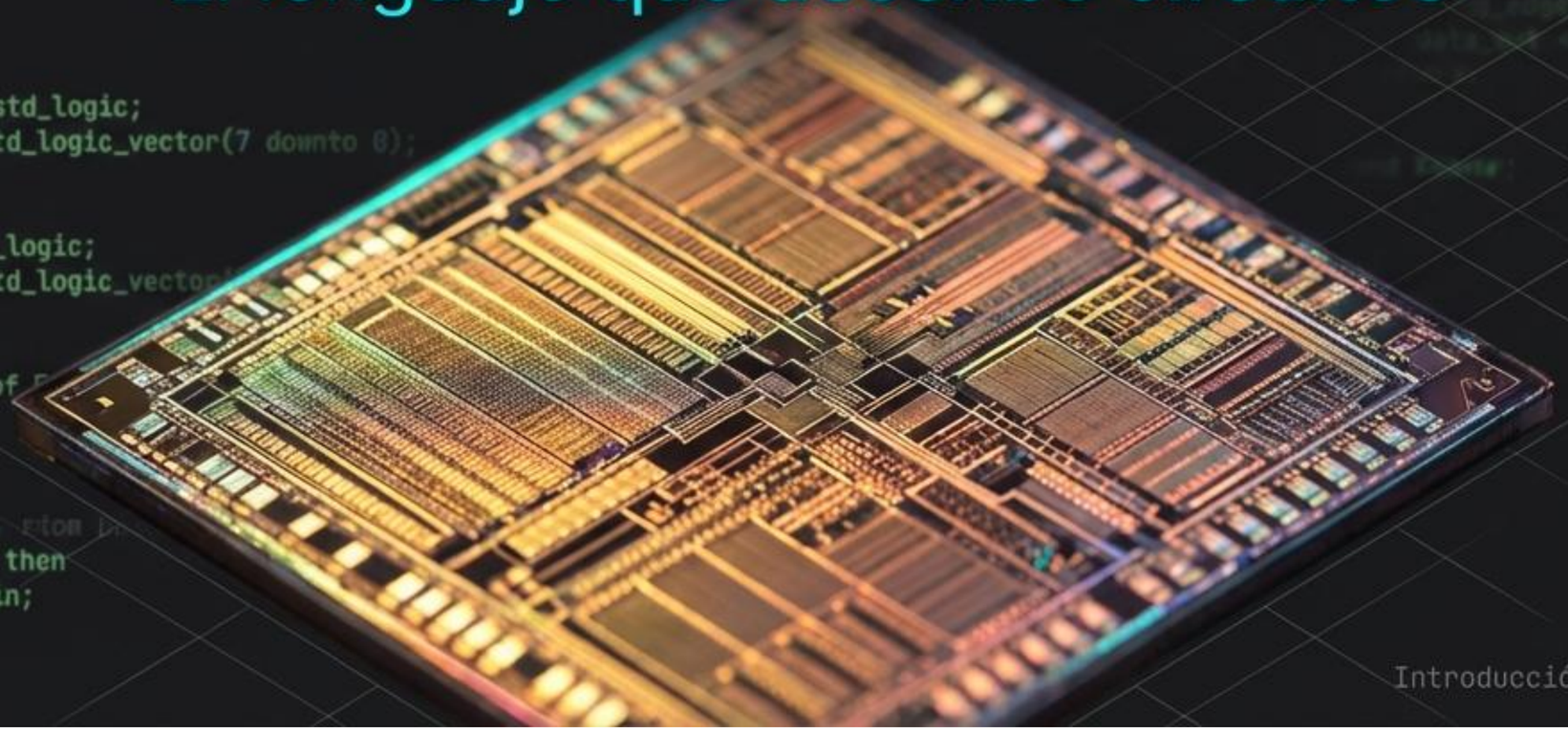
Introducción a VHDL: Pensando en Hardware

El lenguaje que **describe circuitos**



La Trampa del Principiante: ¿Qué NO es VHDL?

NO es secuencial: El código no se ejecuta línea por línea.

NO es software: No creamos programas para Windows/Linux.

NO existe el “después”: En hardware, todo ocurre al mismo tiempo (Concurrencia).

Si piensas en ‘primero A y luego B’, el diseño fallará.

El Poder de VHDL

Documentación

Descripción formal del circuito.

Simulación

Probar en PC antes de quemar.

Síntesis

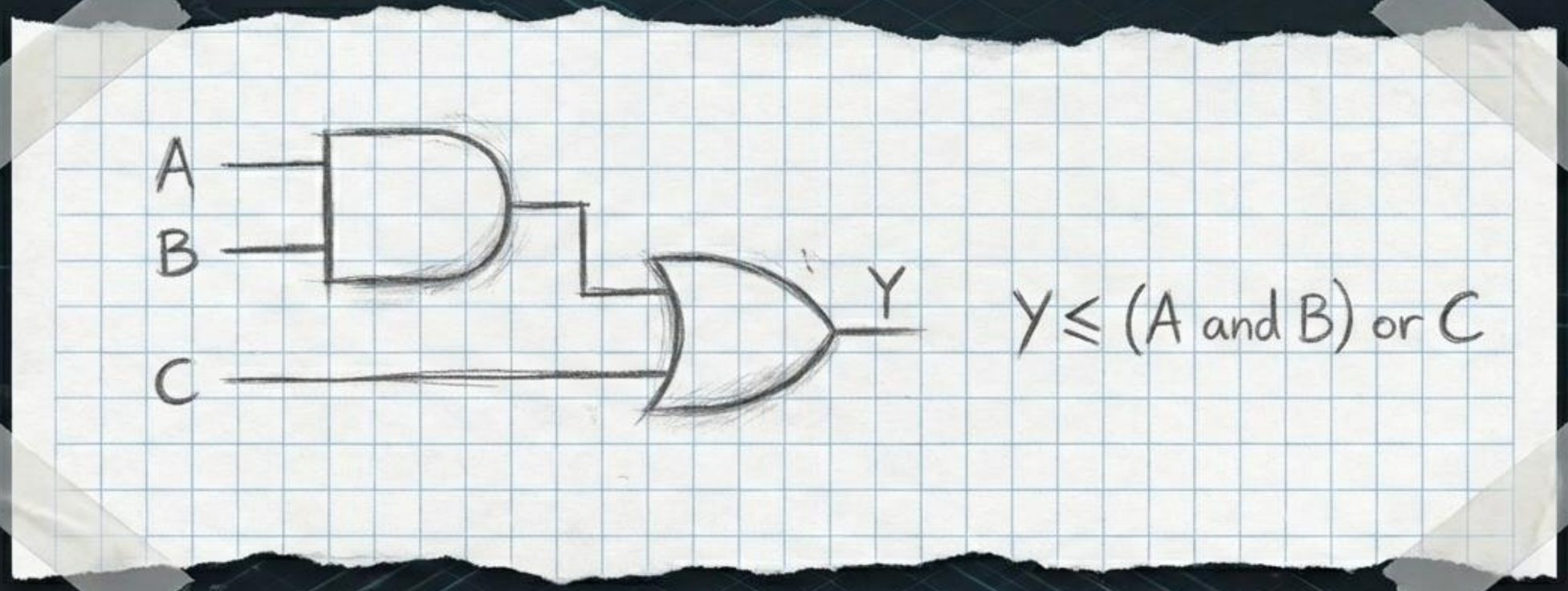
Traducción a hardware real.

Reutilización

Diseña una vez, usa siempre.

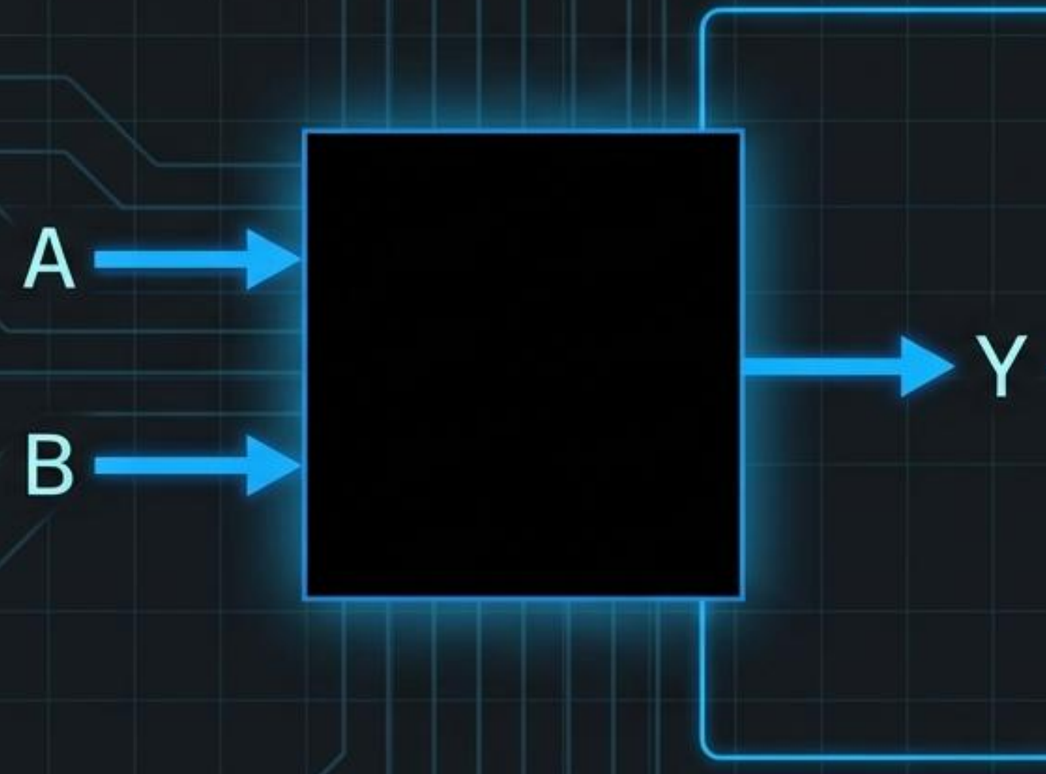
Nuestro Método de Trabajo: El Flujo Ideal

Paso 1: Nunca subestimes el Lápiz



Antes de escribir código, debemos dibujar qué queremos construir. Esto fuerza al cerebro a pensar en hardware.

Paso 2: La Entity (La Caja Negra)



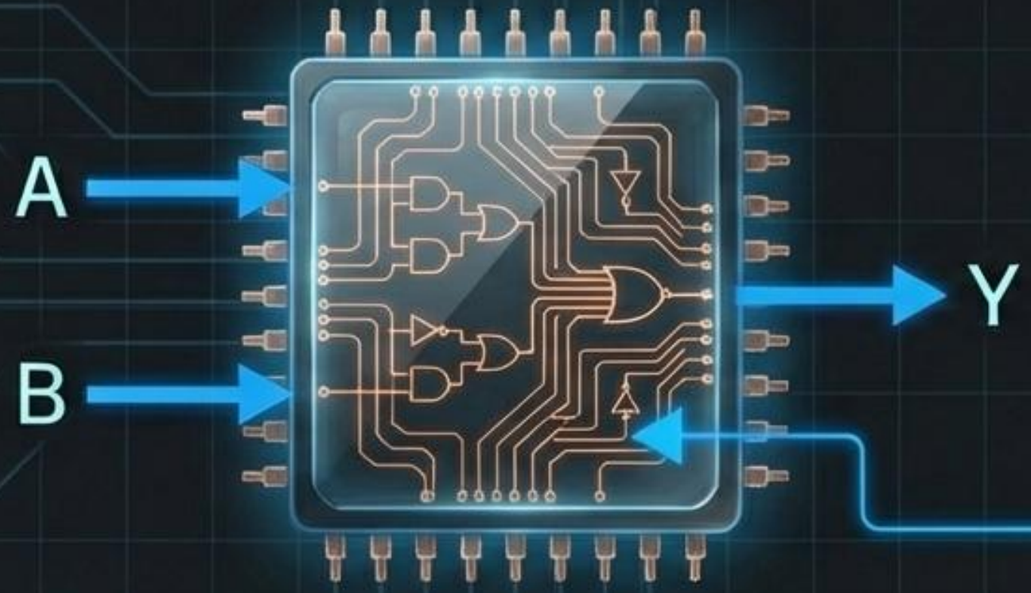

```
entity mi_circuito is
   port (
      A : in bit;    -- Entrada
      B : in bit;    -- Entrada
      Y : out bit    -- Salida
   );
end entity mi_circuito;
```

Tipos de Datos: El Vocabulario

bit	Lo básico. Valores: '0', '1'. Ideal para aprender.
bit_vector	Un bus de datos. Agrupación de bits.
std_logic	El estándar industrial (IEEE 1164). 9 estados ('0', '1', 'Z', 'X'...).

Hoy usaremos 'bit' por simplicidad, pero 'std_logic' es el rey en el mundo profesional.

Paso 3: La Architecture (Las Tripas)




```
architecture behavior of mi_circuito is
  -- Aquí declaramos cables internos (signals)

begin
  -- Aquí describimos el funcionamiento
end architecture behavior;
```

Architecture = CÓMO funciona

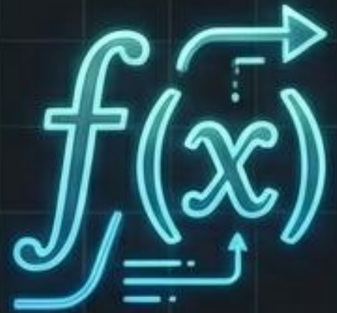
3 Formas de Describir lo Mismo

Ejemplo: Y <= A and B

Estructural

Conecto componentes.
Lista de cables y partes.

Flujo de Datos



Escribo ecuaciones.
Uso ‘and’, ‘or’, ‘not’.
(Nuestro enfoque hoy).

Comportamental

Describo algoritmos.
Parece software
(if/else).

El Diseño Completo: Anatomía del Código

ENTITY (Pines)

```
entity compuerta_and is
    port( a : in bit;
          b : in bit;
          c : out bit
    );
end entity;
```

```
architecture dataflow of compuerta_and is
begin
    c <= a and b; -- Asignación concurrente
end dataflow;
```

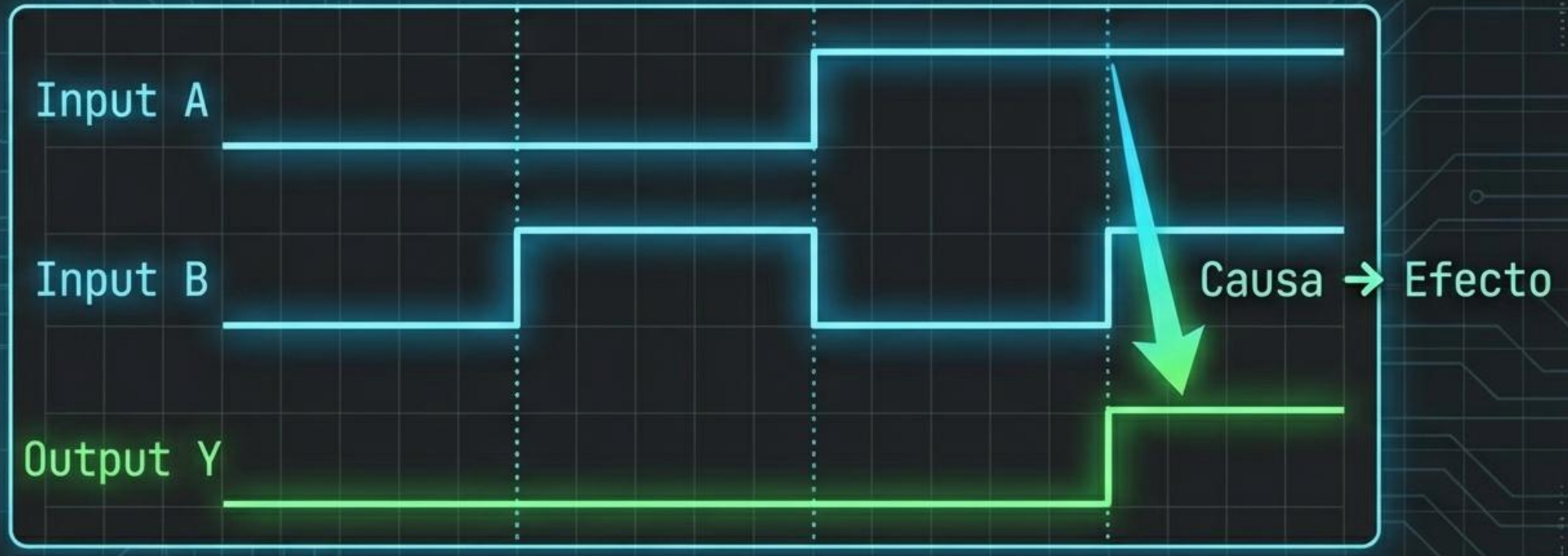
ARCHITECTURE (Lógica)

Paso 4: La Simulación

Verificar antes de Quemar

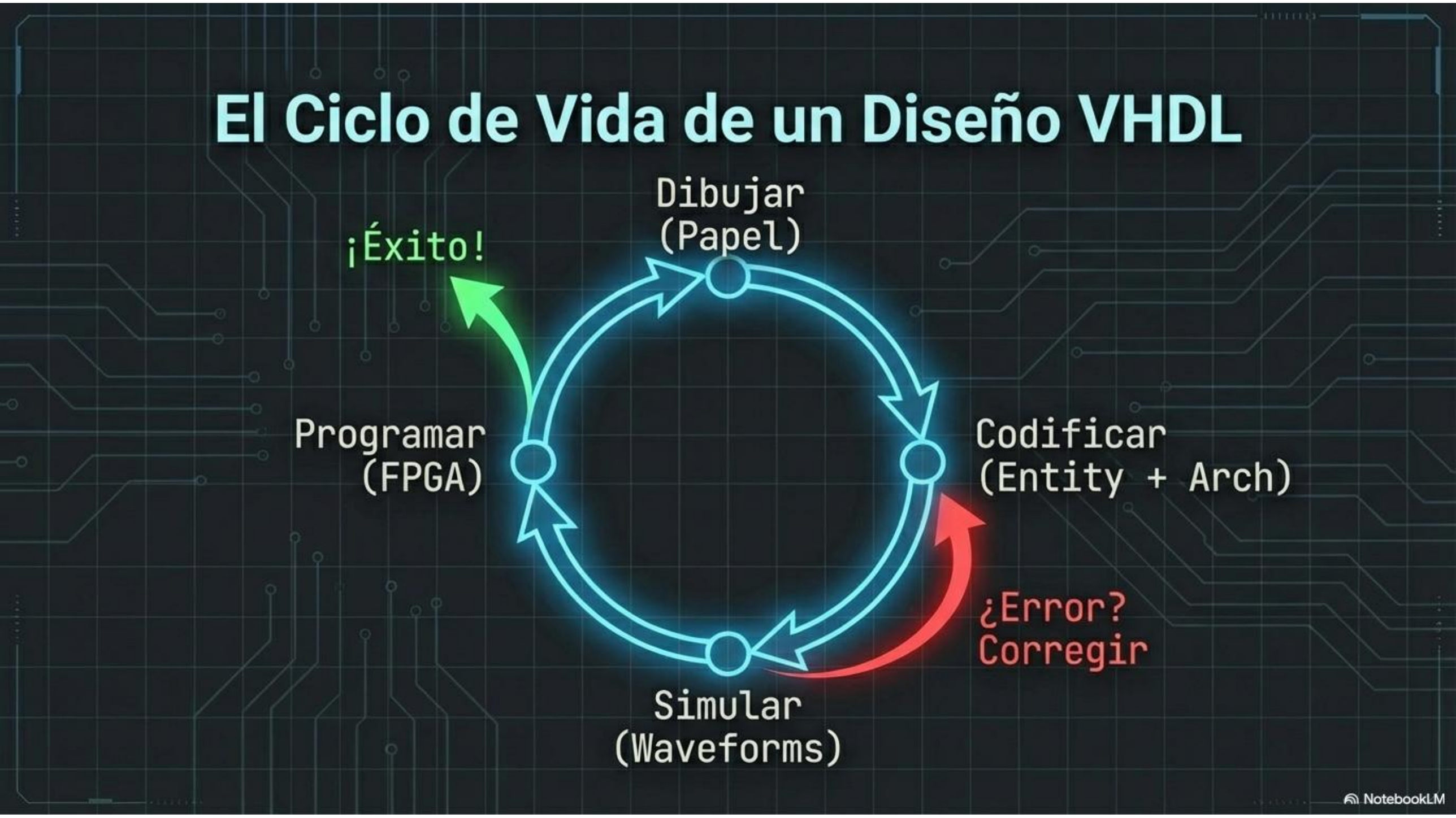
- Enviar a FPGA sin simular es arriesgado.
- Inyectamos valores virtuales (Inputs).
- Observamos el comportamiento (Outputs).

Leyendo el “Waveform”



Y solo se activa cuando A y B están arriba.
¡El código funciona!

El Ciclo de Vida de un Diseño VHDL



Próximos Pasos: Manos a la Obra

1. Abrir Quartus.
2. Escribir el código de la compuerta AND.
3. Simular.
4. RETO: ¡Controlar los LEDs reales!