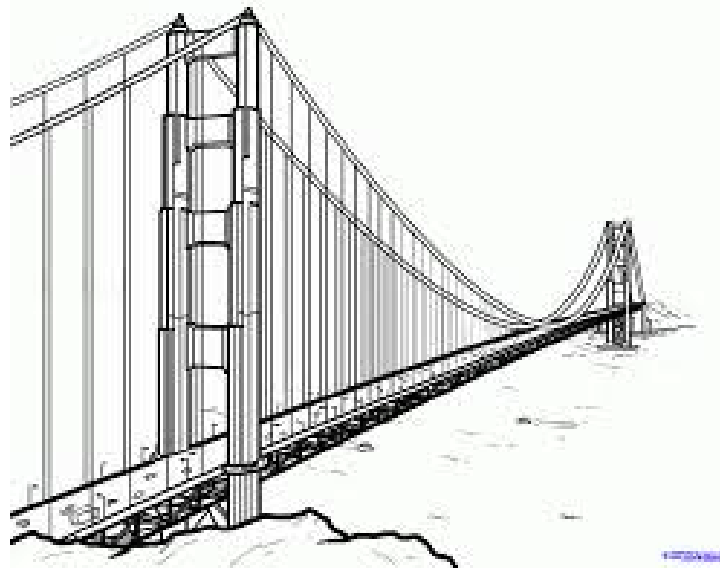


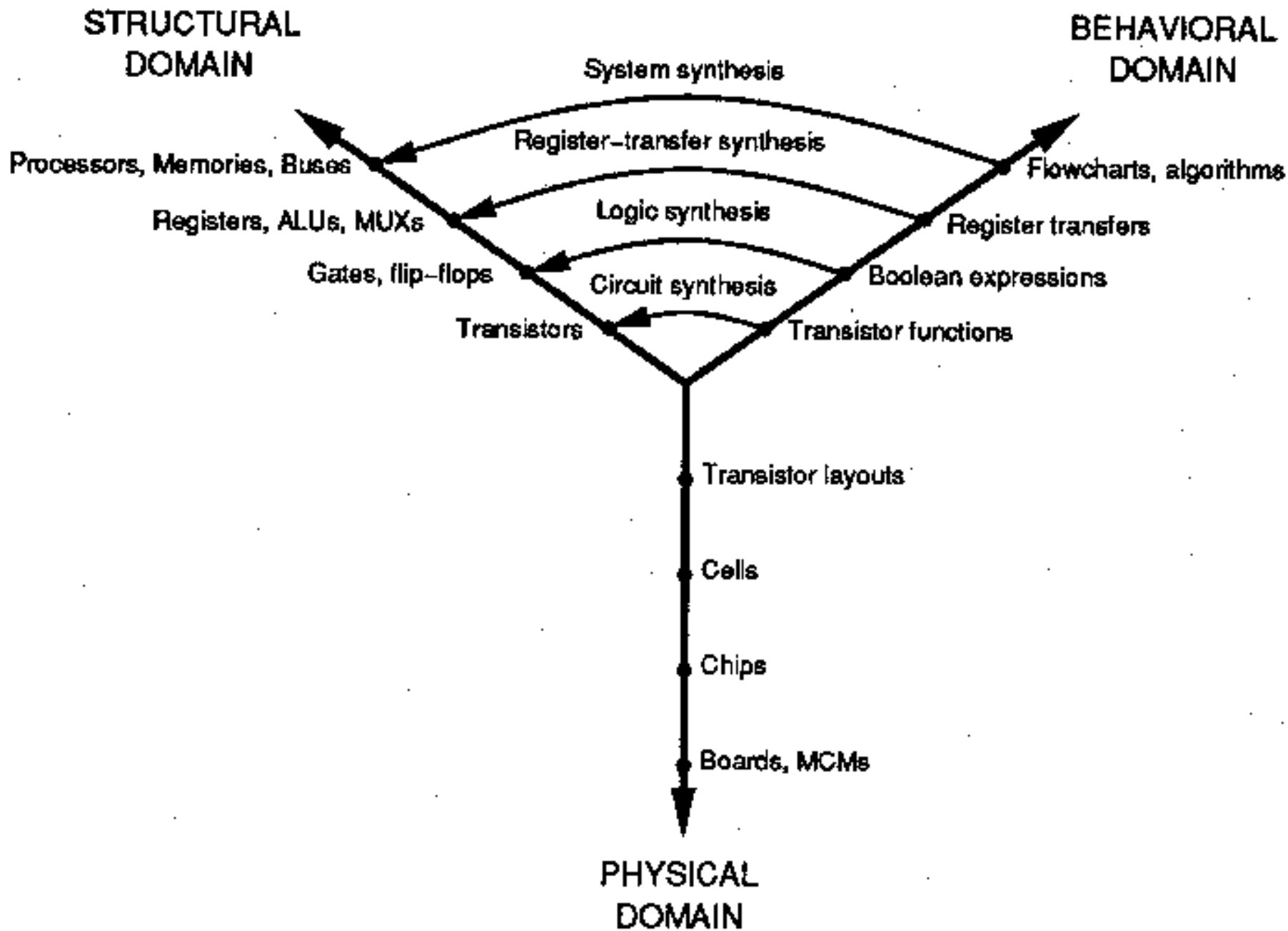
Diseño Lógico 2

Metodología de Diseño

Julio Pérez – Sebastián Fernández
Instituto de Ingeniería Eléctrica
2019

Metodología y herramientas de diseño





Fuente: Gajski et al., "High-Level Synthesis: Introduction to Chip and System Design", 1992

Suggested design methodology

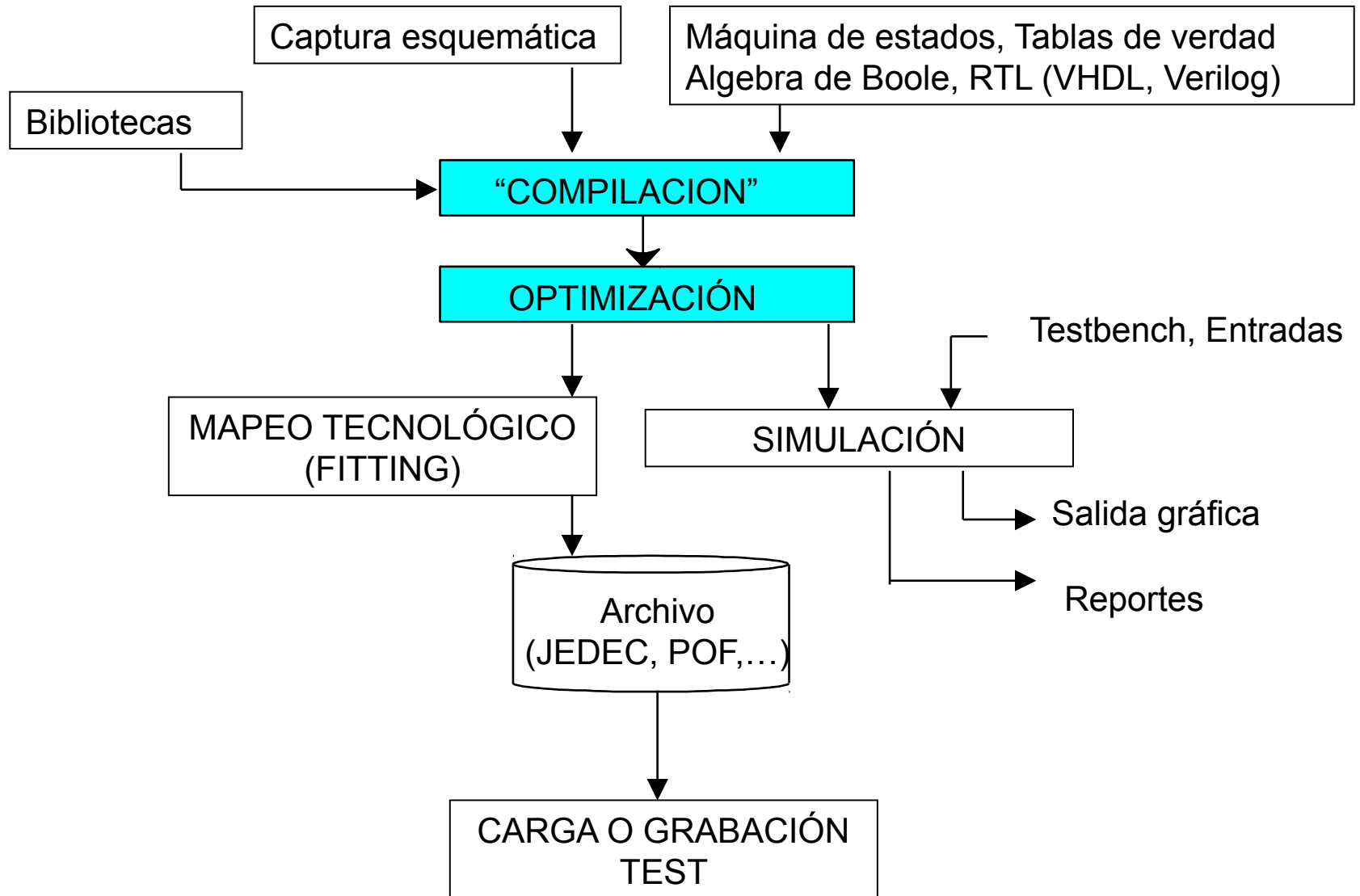
1. Understand the specifications
2. Specify the interface (input and output ports)
3. Identify what data path elements are needed
4. **Draw a block diagram** show the data path elements and interconnect
5. Identify necessary control signals for data path elements
6. Control signals specify the interface of the control path
7. Derive ASM chart(s) for the FSM(s) in the control path
8. Add the control path to block diagram

9. Now (**not before**) describe your design in VHDL. Use a structural approach
10. Simulation, synthesis, implementation...

(de <http://www.sm.luth.se/csee/courses/smd/098>)

Ingreso del diseño

- Captura de esquemáticos
- Tabla de verdad
- Algebra de boole
- Diagrama de estados
- Lenguajes de descripción hardware:
 - » VHDL, Verilog
- Más recientemente: lenguajes de más alto nivel de abstracción (SystemC, SystemVerilog, ...)

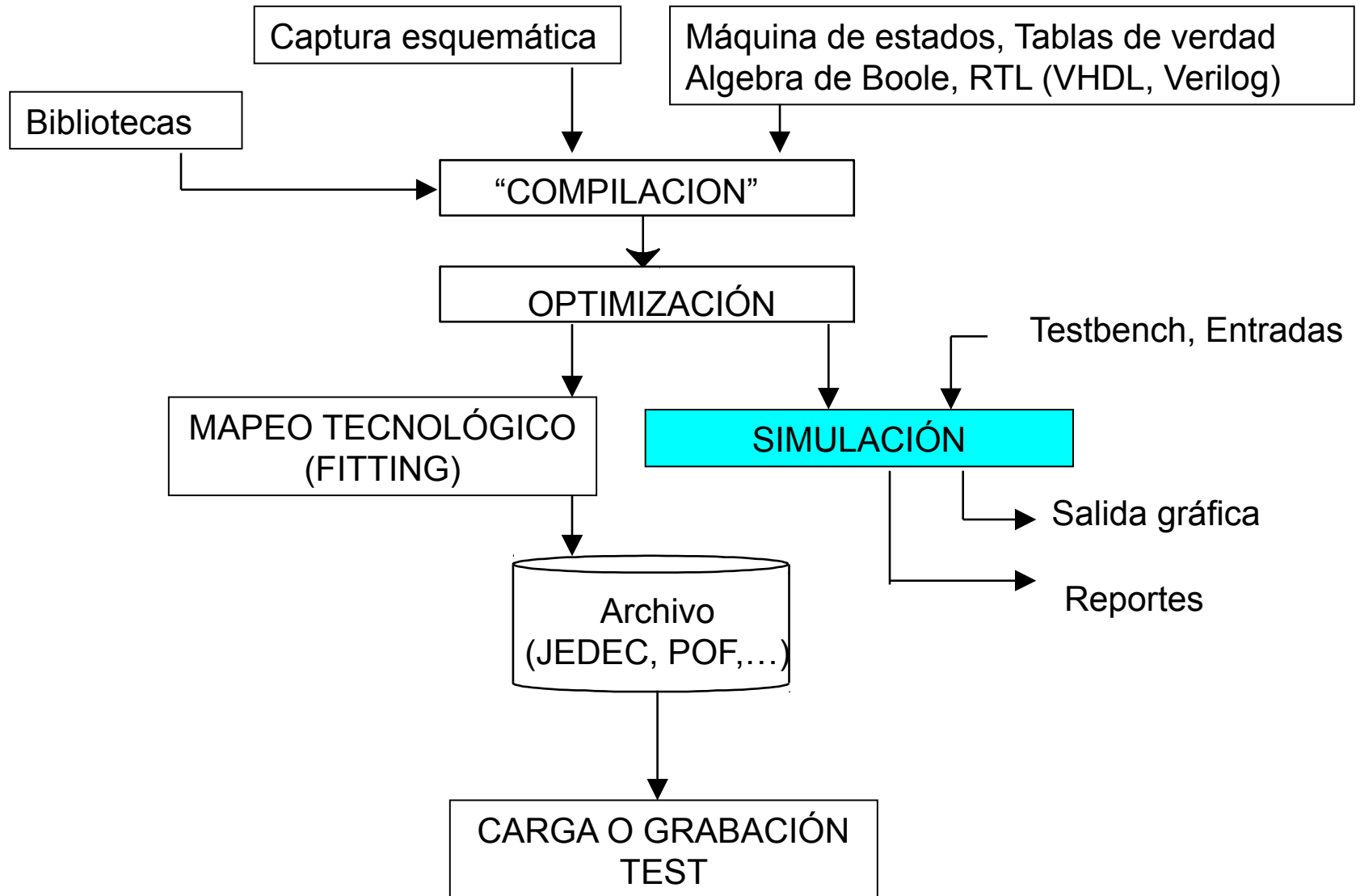


"Compilación"

- Reducción de
 - » descripción "amigable" --> descripción interna de la herramienta
- Inconsistencias
- Errores de sintaxis

Optimización

- Diferentes dimensiones
 - » Area vs. Velocidad vs. Consumo
 - » Respetando restricciones
- Área
 - » Quine-McCluskey, Presto, Espresso
 - » Depende de arquitectura destino
- Experimentar con herramienta para “sacarle jugo”



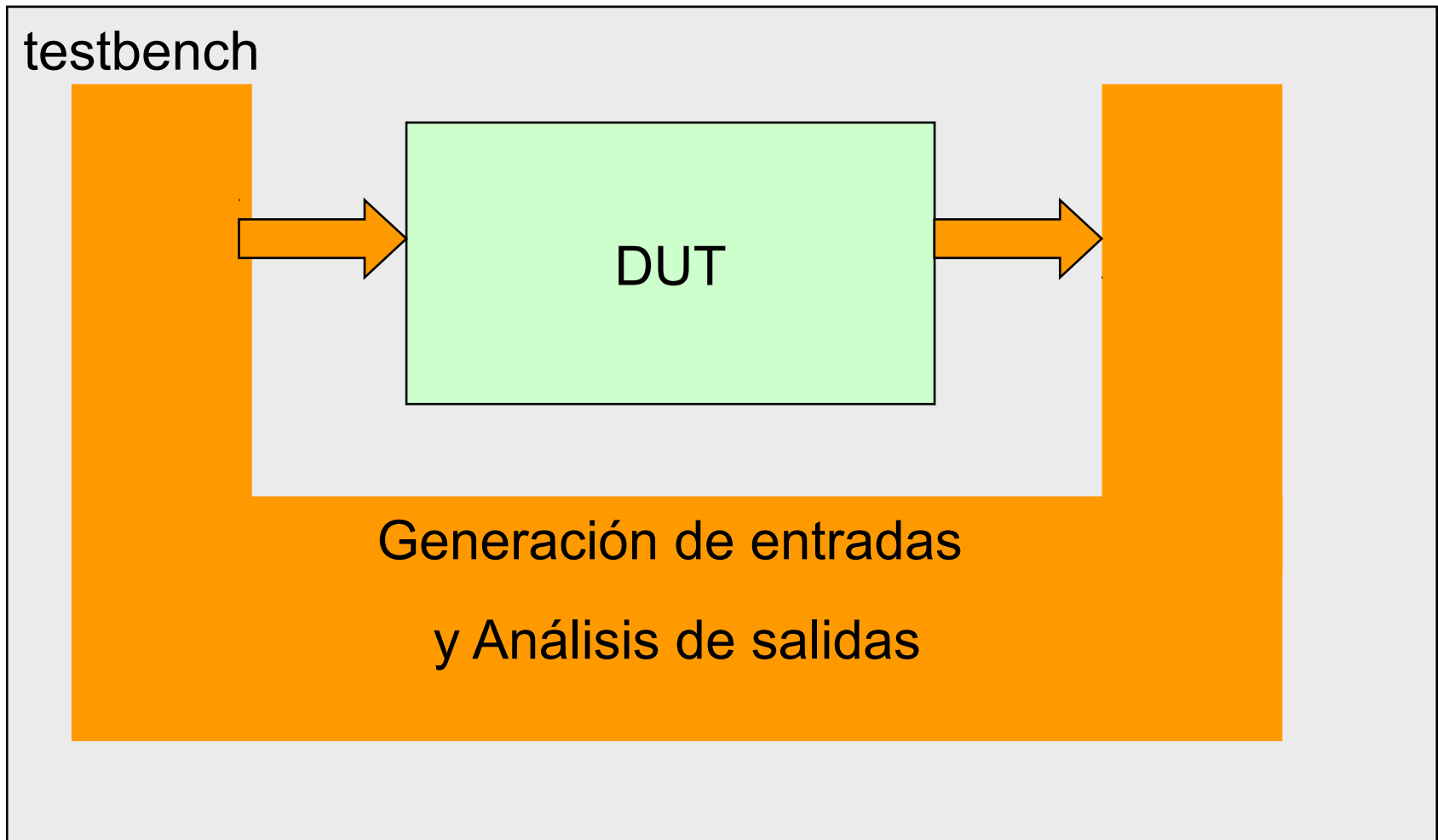
Verificación

- Validación del diseño
 - » Verificación dinámica
 - Corrida sobre un modelo (simulación) o sistema real
 - Estímulos, Análisis de salidas
 - » Verificación estática
 - Comparación de descripciones en diferentes niveles
 - Asignación pines
 - » Verificación de temporización (*Timing Analyzer*)
 - » Verificación de consumo (*PowerPlay*)

Simulación

- ¡Tener claro que espero!
- ¿Por qué no anda si me simuló OK?
 - » Inicialización de registros
 - » Retardos
- Especificación de la simulación:
 - » Vectores de prueba
 - » Dibujo formas de onda entrada (y salida esperada)
 - » Testbench en lenguaje VHDL o Verilog

Testbench



Simulación

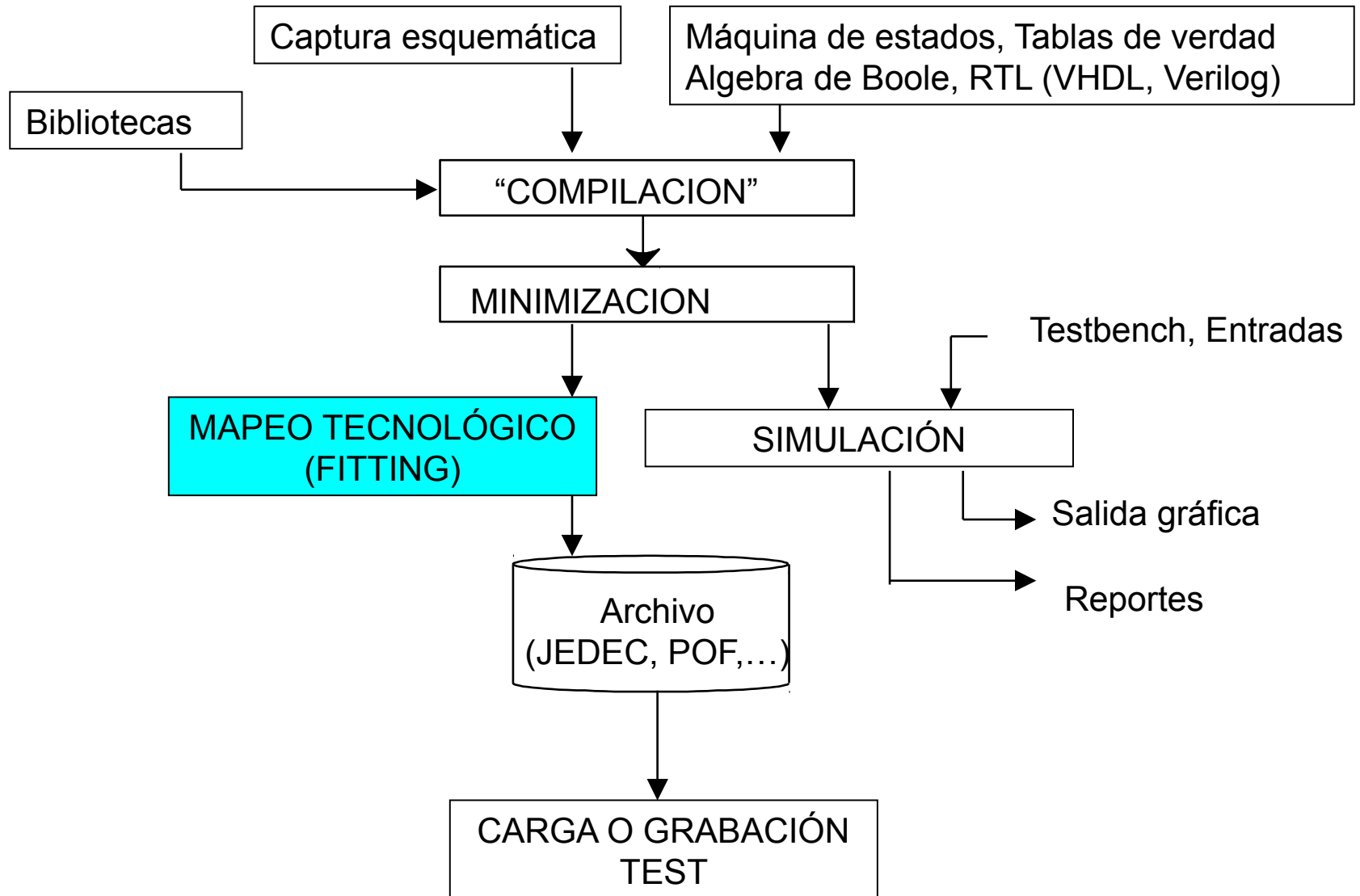
Testbench

- Testbench
 - » entidad sin I/O
 - » una instancia del “Device Under Test”
 - » señales internas para manejar entradas / verificar salidas del DUT
 - » en general no sintetizable

```
Entity probador is
end probador;
architecture test of probador is
  signal sx1, sx2, sy: std_logic
begin
  -- instanciacion de dut
  -- conectando entradas y salidas
  -- ...
  test_seq: process
  begin
    x1 <= '0';  x2 <= '0';
    wait for 10 ns;
    x1 <= '1';  x2 <= '0';
    ...
  end
end
```

Simulación

- Inicialmente se examina el resultado en forma visual
- Es importante poder repetir las simulaciones en modo batch y solo reportar si algún test da mal.
 - » Comparando con corridas anteriores.
 - » Comando *assert*
 - Evalúa una expresión, si es falsa reporta un mensaje.



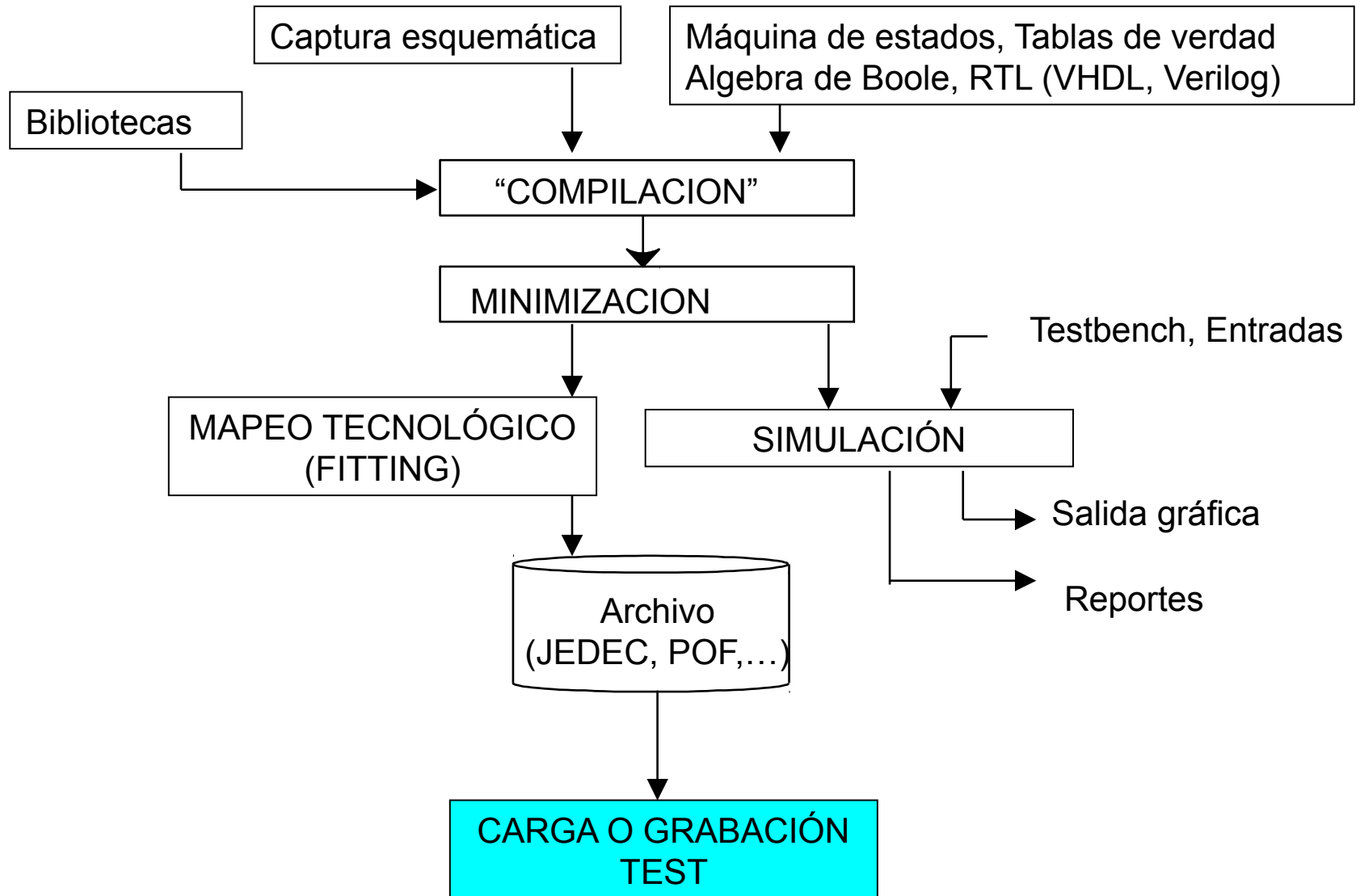
Adaptación a un dispositivo (fitting)

- cabe un diseño dado en un chip dado?
- dado un diseño y criterios, lista ordenada de chips en los que cabe
- partición
 - » minimizar cantidad de chips
 - » minimizar cantidad de señales
 - » retardos
- Si cabe: “place and route”

Adaptación a un dispositivo (fitting)

- Limitaciones
 - » Cantidad de pines
 - » Área (cantidad de celdas lógicas)
 - (o bloques RAM, o multiplicadores, ...)
 - » Conexionado interno

 - » Retardos
 - » Consumo



Grabación

Tecnologías no volátiles

- FUSIBLE
- ANTIFUSIBLE

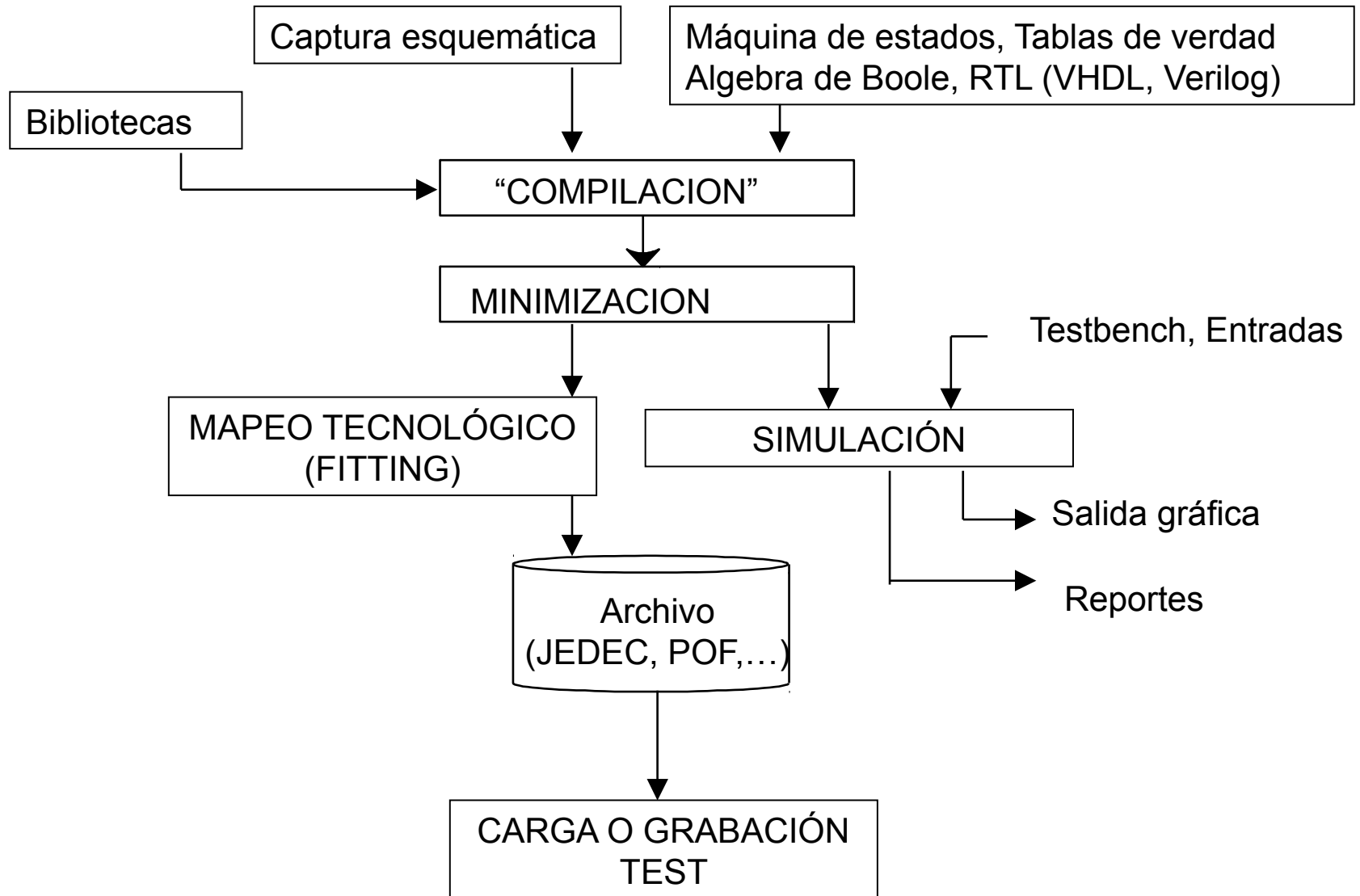
- EPROM
- EEPROM

- Structured Asics (MPLDs, HardCopy)

Grabación

Tecnologías Volátiles

- RAM estática
- Modos de inicialización
 - » Serie o paralelo
 - » Activo o pasivo
 - » JTAG
 - » Procesador en SoC



Prueba

- Controlabilidad, Observabilidad
- Métodos con “cama de clavos”
- Métodos de barrido (boundary scan)
 - » Modo prueba
 - FF conectados en un gran shiftreg (scan chain)
 - Carga de valores iniciales (controlabilidad)
 - Flanco de reloj (modo normal)
 - Extracción de estado final (observabilidad)

Prueba

JTAG IEEE Std 1149.1- 1990

- Método Boundary scan
- Inicialmente para test de placas
- Test Access Port (TAP):
 - » Test Data Input (TDI)
 - » Test Data Output (TDO)
 - » Test Mode Select (TMS)
 - » Test Clock Input (TCK)
 - » Test Reset Input (TRST)

Prueba Built In Self-Test (BIST)

- Cada módulo se auto-testea generando internamente una secuencia pseudoaleatoria de vectores de test
- Se comanda externamente el inicio del test. Se obtiene como resultado un “*hash*” de la secuencia de resultados que externamente se puede comparar con el valor esperado
- La información a transferir se limita al *hash* acortando la duración del test

Herramientas

- "Cerradas" o "abiertas"
- Formatos de intercambio
 - » EDIF
 - Electronic Design Interchange Format
 - » JEDEC
 - » netlists (listas de conexiones) varios formatos, inclusive VHDL (estructural)
 - » propietarios

Suggested design methodology

1. Understand the specifications
2. Specify the interface (input and output ports)
3. Identify what data path elements are needed
4. **Draw a block diagram** show the data path elements and interconnect
5. Identify necessary control signals for data path elements
6. Control signals specify the interface of the control path
7. Derive ASM chart(s) for the FSM(s) in the control path
8. Add the control path to block diagram

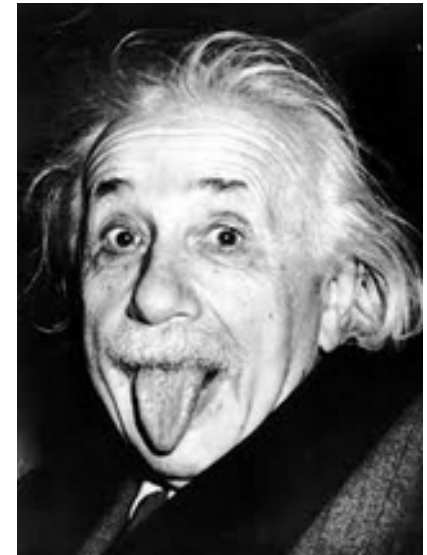
9. Now (**not before**) describe your design in VHDL. Use a structural approach
10. Simulation, synthesis, implementation...

(de <http://www.sm.luth.se/csee/courses/smd/098>)

Diferencia: pasos 1 a 8



VS



Restricciones

- Área (de silicio)
 - » Celdas, memoria, recursos en general
- Temporización
 - » Retardos, frecuencia máxima
- Consumo
 - » Potencia, energía
- Costo
 - » \$\$\$