**🌟 Giai đoạn 1: Làm quen với Verilog**

**🔹 Mục tiêu:**

* Hiểu cú pháp cơ bản
* Viết mô-đun đơn giản
* Làm quen với mô phỏng

**📘 Nội dung:**

1. **Giới thiệu Verilog**
   * HDL là gì? (Hardware Description Language)
   * So sánh Verilog vs VHDL
   * Dùng Verilog để mô tả mạch số như thế nào?
2. **Cài đặt môi trường**
   * Phần mềm mô phỏng: ModelSim / Vivado / Icarus Verilog
   * Dùng Quartus (Intel FPGA) hoặc Vivado (Xilinx FPGA)
3. **Cấu trúc cơ bản của Verilog**
   * module, endmodule
   * input, output, wire, reg
   * assign, always, initial
   * Cách viết testbench
4. **Viết các mô-đun đơn giản**
   * Cổng logic (AND, OR, XOR…)
   * Mạch cộng 1 bit (half-adder, full-adder)
   * Mạch giải mã, encoder, multiplexer

**🧠 Giai đoạn 2: Kỹ thuật mô tả mạch số**

**🔹 Mục tiêu:**

* Mô tả được mạch tổ hợp và tuần tự
* Biết phân biệt blocking vs non-blocking (= vs <=)
* Viết các mô hình FSM

**📘 Nội dung:**

1. **Mạch tổ hợp (combinational)**
   * Cách viết assign
   * Sử dụng always @(\*)
   * Các mạch: MUX, demux, comparator, ALU
2. **Mạch tuần tự (sequential)**
   * Cách viết always @(posedge clk)
   * Định nghĩa reg và wire
   * Flip-Flop, thanh ghi, bộ đếm (counter)
3. **Finite State Machine (FSM)**
   * Mô hình Moore và Mealy
   * Thiết kế máy bán hàng (vending machine), traffic light…

**⚙️ Giai đoạn 3: Mô phỏng và kiểm thử**

**🔹 Mục tiêu:**

* Viết testbench để kiểm thử
* Sử dụng waveform
* Debug lỗi logic

**📘 Nội dung:**

1. Viết Testbench đơn giản
2. Hiểu initial, #delay, $display, $monitor
3. Dùng waveform để kiểm tra tín hiệu
4. Tạo file .vcd để xem sóng (với Icarus Verilog)

**🛠️ Giai đoạn 4: Dự án trên FPGA**

**🔹 Mục tiêu:**

* Viết và chạy code thực tế trên FPGA
* Sử dụng LED, switch, màn hình 7 đoạn
* Hiểu cách tổng hợp (synthesis) và triển khai (implementation)

**📘 Nội dung:**

1. Làm quen phần cứng FPGA (DE2-115, DE10-Lite, PYNQ-Z2…)
2. Thiết kế:
   * Mạch nhấn nút – bật LED
   * Counter hiển thị trên LED 7 đoạn
   * MUX chọn đầu vào
   * FSM điều khiển
3. Ánh xạ chân (pin assignment)
4. Tạo sơ đồ thời gian (timing diagram)

**🚀 Giai đoạn 5: Nâng cao & Tối ưu**

**🔹 Mục tiêu:**

* Viết code hiệu quả
* Tối ưu logic
* Biết tạo IP Core

**📘 Nội dung:**

1. Tối ưu về timing, logic
2. Viết Verilog sinh tham số (parameter, generate)
3. Thiết kế RAM, ROM
4. Giao tiếp:
   * UART, SPI, I2C
   * Giao tiếp cảm biến, LCD, servo
5. Tạo IP Core dùng trong hệ thống lớn

**🔁 Tài nguyên học tập**

* **Sách**:
  + *Verilog HDL* – Samir Palnitkar (Rất phổ biến, dễ hiểu)
  + *FPGA Prototyping by Verilog Examples* – Pong P. Chu
* **Video**: FPGA4Student, HDLBits, YouTube: EEVblog
* **Website thực hành**:
  + [HDLBits](https://hdlbits.01xz.net/) – Luyện tập trực tuyến
  + [EDA Playground](https://www.edaplayground.com/) – Mô phỏng Verilog online

**✅ Gợi ý lộ trình theo tuần (có thể điều chỉnh)**

| **Tuần** | **Mục tiêu chính** |
| --- | --- |
| 1-2 | Cú pháp cơ bản, mạch tổ hợp |
| 3-4 | Mạch tuần tự, FSM, testbench |
| 5-6 | Làm quen FPGA, chạy LED, Counter |
| 7-8 | Dự án nhỏ: Traffic Light, Vending Machine |
| 9+ | Giao tiếp ngoại vi, tối ưu hóa, dự án nâng cao |