Lab 2: Mạch logic dãy và ứng dụng

Yêu cầu báo cáo:

Trong báo cáo nên có các phần sau:

- Source code VHDL; kịch bản kiểm tra (dưới dạng code VHDL hoặc lệnh của Modelsim) của các phần thực hành (những phần phải viết code)
- Kết quả chạy mô phỏng, giải thích kết quả (dưới dạng dễ hiểu và dễ thấy)
- Trả lời các câu hỏi trong phần hướng dẫn
- Liên kết đến các phần lý thuyết đã học và những kiến thức mà các bạn đã tìm hiểu để làm bài thực hành.

Chuẩn bị

Tạo thư mục và chuyển đến thư mục làm việc bằng cách dùng lệnh:

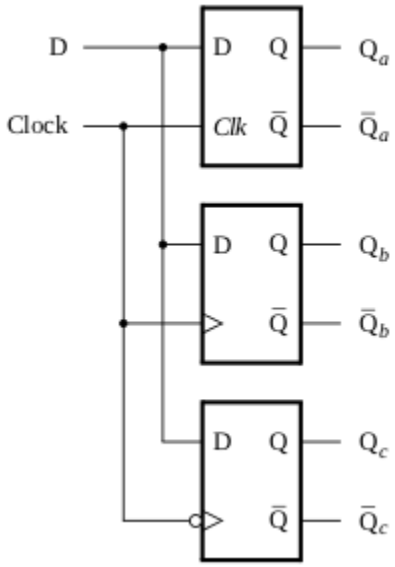
```
mkdir -p $HOME/icdesign/m2
cp -a /home/tools/synopsys/m2/lab2 $HOME/icdesign/m2
cd $HOME/icdesign/m2/lab2
```

Cấu trúc cây thư mục

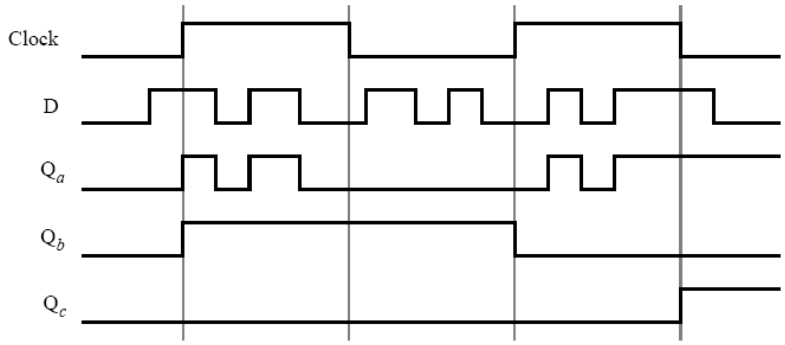
```
.
├── sim
│   ├── Makefile
│   └── scripts
│       ├── accumulator_tb.tcl
│       ├── flipflops_tb.tcl
│       └── mac_tb.tcl
├── src
│   ├── accumulator.vhd
│   ├── flipflops.vhd
│   └── mac.vhd
└── tb
    ├── accumulator_tb.vhd
    ├── flipflops_tb.vhd
    └── mac_tb.vhd
```

Phần 1: Các loại flip-flop

Cho mạch điện như trong Hình 1. Hình 2 thể hiện quan hệ giữa input và output của mạch điện này với các đầu ra Qa, Qb, Qc.



Hình 1. Mạch điện của ba phần tử bộ nhớ khác nhau.



Hình 2. Lược đồ thời gian của mạch điện trong Hình 1.

1. Hoàn thành kịch bản kiểm tra của các flip-flop ở Hình 1 trong tập tin flipflops_tb.vhd bằng cách tạo ra dạng sóng của tín hiệu *d* và tín hiệu *clk*; biên dịch và chạy mô phỏng thiết kế dùng công cụ VCS và Verdi.

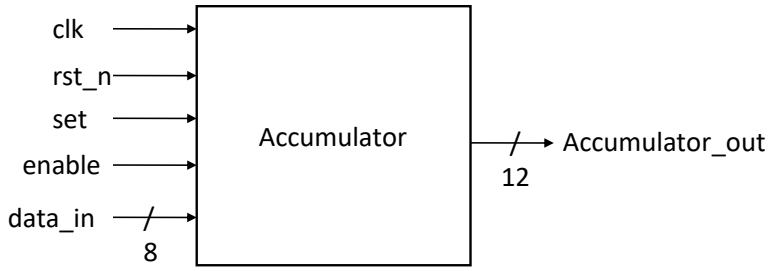
```
vhdlan -nc -kdb ../src/flipflops.vhd
vhdlan -nc -kdb ../tb/flipflops_tb.vhd
vcs -nc -kdb -debug_access+all flipflops_tb
./simv -ucli -do scripts/flipflops_tb.tcl
verdi -ssf flipflops_tb.fsdb
```

2. Nhận xét về sự khác nhau trong tín hiệu đầu ra theo quan hệ với tín hiệu đầu vào và tín hiệu xung nhịp đồng hồ clk. Các phần tử trong mạch điện ở Hình 1 thuộc loại nào? Gọi tên các mạch điện đó.
3. Đọc code của tập tin flipflops.vhd từ đó trả lời các câu hỏi sau:
    a. Danh sách nhậy cảm (sensitivity list) là gì?
    b. Khi nào thì một process được chạy trong ngôn ngữ VHDL. Nêu ứng dụng của process trong VHDL để miêu tả các mạch điện tổ hợp và các mạch tuần tự.

Phần 2: Mô hình hóa và mô phỏng bộ cộng tích lũy (Accumulator)

Trong Hình 3 là các đầu vào và đầu ra của một bộ cộng tích lỹ bao gồm các tín hiệu:

STT	Tên tín hiệu	Chế độ	Độ rộng bit	Miêu tả
1	Clk	In	1	Tín hiệu xung nhịp đồng hồ của bộ cộng tích lũy. Đầu ra của bộ đếm thay đổi ở sườn lên của tín hiệu clk.
2	Rst_n	In	1	Tín hiệu reset. - Khi rst_n = '0' bộ cộng tích lũy ở chế độ reset. - Khi rst_n = '1', bộ cộng tích lũy ở chế độ hoạt động.
3	Set	In	1	- Khi *set* = '1', giá trị đầu ra của bộ cộng tích lũy bằng data_in. - Khi *set* = '0', các chế độ hoạt động khác.
4	data_in	In	8	Giá trị được nạp vào các thanh ghi của bộ đếm khi set = '1'
5	enable	In	1	Tín hiệu cho phép cộng tích lũy - Khi *enable* = '1' giá trị của thanh ghi R được cộng thêm 1 lượng data_in sau mỗi sườn lên của tín hiệu clk: R = R + data_in - Khi *enable* = '0' và các tín hiệu khác ở chế độ không kích hoạt, các thanh ghi giữ nguyên giá trị.
6	accumulator_out	Out	12	Giá trị đầu ra của bộ đếm



Hình 3. Các chân vào ra của bộ cộng tích lũy 8 bit đầu vào và 12-bit đầu ra.

Trong phần này, chúng ta sẽ mô hình hóa một bộ đếm tiến sử dụng ngôn ngữ VHDL thực hiện các chức năng như đã được miêu tả trong phần trên.
Gợi ý:
Có thể dùng các phép cộng bằng cách dùng kiểu dữ liệu unsigned/signed trong package numeric_std của thư viện IEEE.

1. Mô hình hóa hành vi của bộ cộng tích lũy được miêu tả ở trên dùng ngôn ngữ VHDL. Tạo kịch bản kiểm tra cho bộ cộng tích lũy này, chạy mô phỏng và kiểm tra lại kết quả mô phỏng.

Gợi ý: có thể dùng lệnh *force* hoặc viết *testbench*.

2. Các tín hiệu điều khiển rst_n, set trong thiết kế của các bạn là tín hiệu điều khiển đồng bộ (synchronous) hay là bất đồng bộ (asynchronous). Viết miêu tả của các tín hiệu này theo cách khác (đồng bộ hoặc bất đồng bộ).
3. Bonus: Kết quả của phép cộng tích lũy thay đổi như thế nào nếu chúng ta thay đổi tín hiệu accumulator với độ rộng bit là 8 bít. Hiện tượng gì sẽ xảy ra và nêu tầm quan trọng của hiện tượng này trong lập trình các phép toán đại số.

Phần 3: Ứng dụng bộ cộng tích lũy và bộ đếm để tính tích chập MAC

Trong phần này, chúng ta sẽ thiết kế một bộ tính tích chập (Multiply & Accumulator (MAC)). Bộ MAC nhận đầu vào là 2 cặp số 8 bit có dấu và sẽ đưa ra đầu ra 18-bit dữ liệu có dấu mỗi khi nhận đủ 4 cặp giá trị đầu vào, bộ MAC sẽ đưa ra kết quả là tổng của các tích của cặp số này.
Cụ thể như sau: giả sử ta có 2 ma trận:

5	7
6	8

A

3	2
1	4

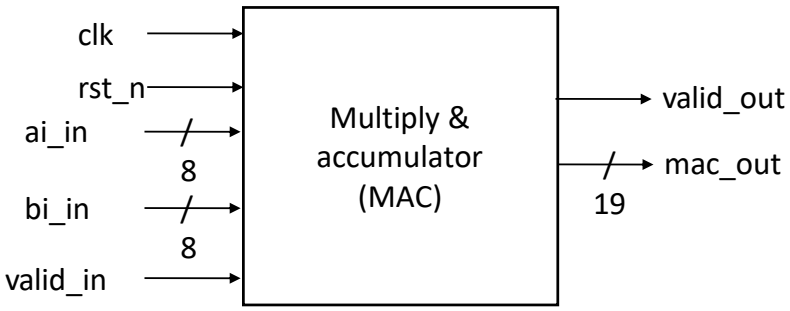
B

$$m = \sum_{k=0}^{n} A_i B_i = 5 * 3 + 7 * 2 + 6 * 1 + 8 * 4 = 67$$

Các tín hiệu vào ra của bộ MAC được miêu tả như sau:

STT	Tên tín hiệu	Chế độ	Độ rộng (bit)	Miêu tả
1	Clk	In	1	Tín hiệu xung nhịp của cả hệ thống
2	Rst_n	In	1	Tín hiệu reset. - Khi *rst_n* = '0', mạch điện ở chế độ reset, các thanh ghi được nạp giá trị mặc định. - Khi *rst_n* = '1', mạch điện ở chế độ hoạt động
3	ai_in	In	8	Tín hiệu đầu vào cho ma trận A
4	bi_in	In	8	Tín hiệu đầu vào hco ma trận B
4	Valid_in	In	1	Tín hiệu thông báo tín hiện đầu vào *ai_in* và *bi_in* có chứa dữ liệu. - Nếu *valid_in* = '1' có dữ liệu đầu vào cần thực hiện tính r = r+ ai*bi. - Nếu *valid_in* = '0', không có dữ liệu đầu vào giữ nguyên kết quả.
6	Valid_out	Out	1	Tín hiệu thông báo tín hiệu đầu ra có chứa dữ liệu. - *valid_out* = '1' mỗi khi có 4 cặp giá trị ai và bi.

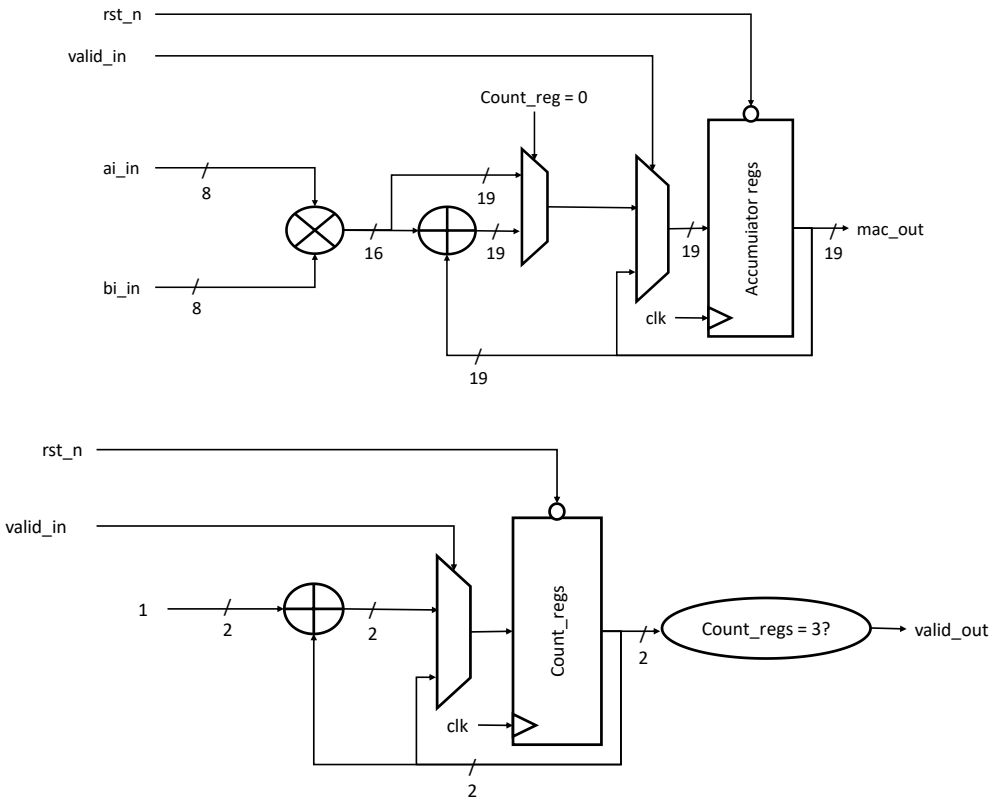
				- *valid_out* = ’0’, khi chưa có đủ dữ liệu và không có kết quả tính toán.
	mac_out	Out	19	Giá trị đầu ra m là tổng của các tích ai*bi



Hình 4. Các chân vào ra của bộ MAC.

1. Mô hình hóa hành vi của bộ MAC dùng ngôn ngữ VHDL. Viết kịch bản kiểm tra cho bộ MAC.
2. Tín hiệu *rst_n* có vai trò gì trong thiết kế này.
3. **Bonus**: Thiết kế bộ MAC cho đầu vào với n-bit với n là 1 số nguyên dương có thể thay đổi tại thời điểm sử dụng thiết kế.

Gợi ý:
Sơ đồ khối của bộ MAC được trình bày trong Hình 5.



Hình 5. Sơ đồ tính toán của bộ MAC.