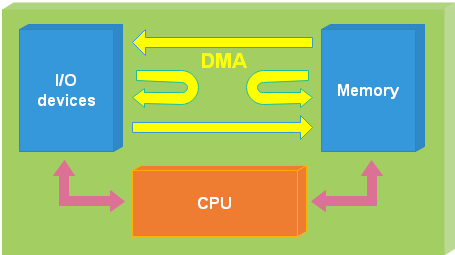
DMA.

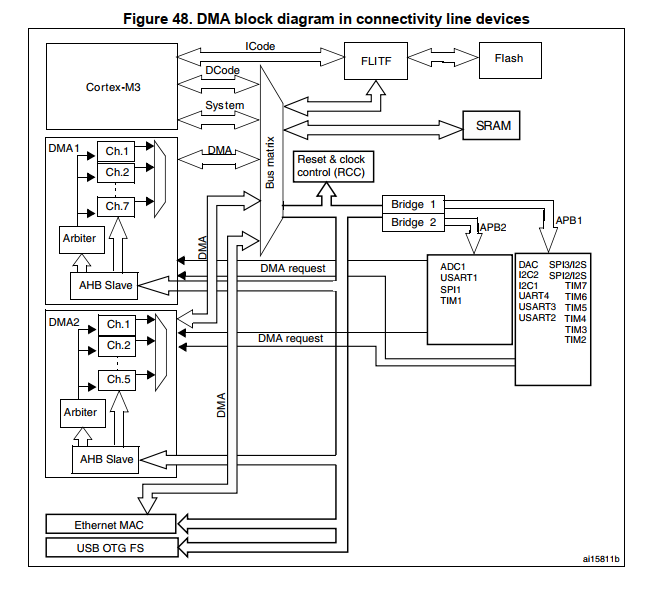
DMA – Direct memory access được sử dụng với mục đích truyền data với tốc độ cao từ thiết bị ngoại vi đến bộ nhớ cũng như từ bộ nhớ đến bộ nhớ.

DMA có thể điều khiển data truyền từ SRAM đến Peripheral - UART và ngược lại, mà không thông qua data bus của CPU.

Với DMA, dữ liệu sẽ được truyền đi nhanh chóng mà không cần đến bất kỳ sự tác động nào của CPU. Điều này sẽ giữ cho tài nguyên của CPU được rảnh rỗi cho các thao tác khác. Đồng thời tránh việc data nhận về từ ngoại vi bị mất mát.

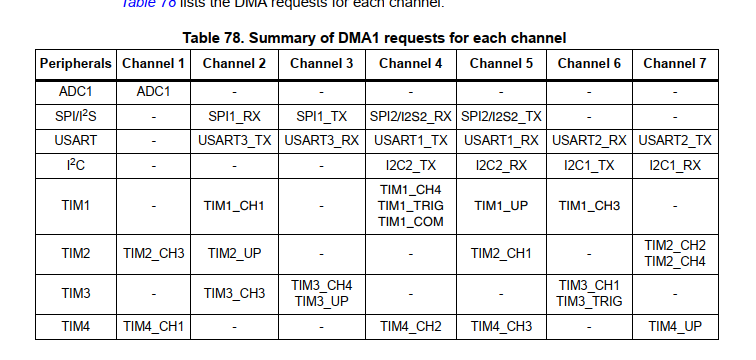


DMA hoạt động không liên quan đến quá trình thực thi của CPU, nhờ đó nó có thể truyền nhận dữ liệu mọi lúc. Chỉ cần có tín hiệu truyền đến, DMA sẽ nhận và lưu vào vùng nhớ cố định. Người dùng chỉ cần truy cập đến vùng nhớ này để lấy dữ liệu cần thiết.



# Sơ đồ bộ DMA trong STM32F1

STM32F1 có 2 bộ DMA với nhiều kênh, mỗi kênh có nhiều ngoại vi có thể dùng DMA như bảng:



* Các Channel đều có thể được cấu hình riêng biệt.
* Mỗi Channel được kết nối để dành riêng cho tín hiệu DMA từ các thiết bị ngoại vi hoặc tín hiệu từ bên trong MCU.
* Có 4 mức ưu tiên có thể lập trình cho mỗi Channel.
* Kích thước data được sử dụng là 1 Byte, 2 Byte (Half Word) hoặc 4byte (Word)
* Hỗ trợ việc lặp lại liên tục Data.
* 5 cờ báo ngắt (DMA Half Transfer, DMA Transfer complete, DMA Transfer Error, DMA FIFO Error, Direct Mode Error).
* Quyền truy cập tới Flash, SRAM, APB1, APB2, APB.
* Số lượng data có thể lập trình được lên tới 65535.
* Đối với DMA2, mỗi luồng đều hỗ trợ để chuyển dữ liệu từ bộ nhớ đến bộ nhớ.

DMA có 2 chế độ hoạt động là normal và circular:

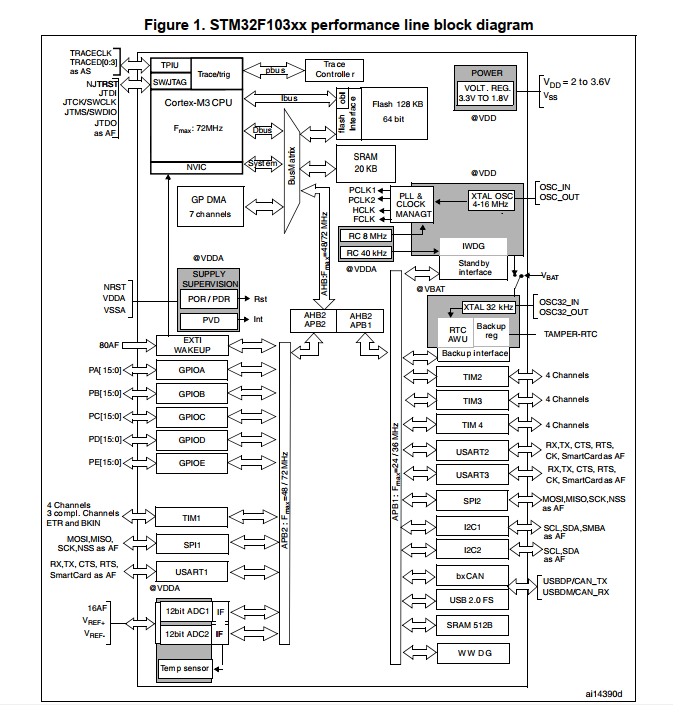
· Normal mode: Với chế độ này, DMA truyền dữ liệu cho tới khi truyền đủ 1 lượng dữ liệu giới hạn đã khai báo DMA sẽ dừng hoạt động. Muốn nó tiếp tục hoạt động thì phải khởi động lại

· Circular mode: Với chế độ này, Khi DMA truyền đủ 1 lượng dữ liệu giới hạn đã khai báo thì nó sẽ truyền tiếp về vị trí ban đầu (Cơ chế như Ring buffer).

# Cấu hình DMA.

DMA truyền nhận dữ liệu từ ngoại vi và bộ nhớ, nên trước hết cần cấp xung và cấu hình tham số cho ngoại vi cần sử dụng DMA.

Không như các ngoại vi khác, DMA cần được cấp xung từ AHB, cả 2 bộ DMA đều có xung cấp từ AHB.



Các tham số cho bộ DMA được cấu hình trong struct DMA\_InitTypeDef. Gồm:

* DMA\_PeripheralBaseAddr: Cấu hình địa chỉ của ngoại vi cho DMA. Đây là địa chỉ mà DMA sẽ lấy data hoặc truyền data tới cho ngoại vi.
* DMA\_MemoryBaseAddr: Cấu hình địa chỉ vùng nhớ cần ghi/ đọc data .
* DMA\_DIR: Cấu hình hướng truyền DMA, từ ngoại vi tới vùng nhớ hay từ vùng nhớ tới ngoại vi.
* DMA\_BufferSize: Cấu hình kích cỡ buffer. Số lượng dữ liệu muốn gửi/nhận qua DMA.
* DMA\_PeripheralInc: Cấu hình địa chỉ ngoại vi có tăng sau khi truyền DMA hay không.
* DMA\_MemoryInc: Cấu hình địa chỉ bộ nhớ có tăng lên sau khi truyền DMA hay không.
* DMA\_PeripheralDataSize: Cấu hình độ lớn data của ngoại vi.
* DMA\_MemoryDataSize: Cấu hình độ lớn data của bộ nhớ.
* DMA\_Mode: Cấu hình mode hoạt động.
* DMA\_Priority: Cấu hình độ ưu tiên cho kênh DMA.
* DMA\_M2M: Cấu hình sử dụng truyền từ bộ nhớ đếm bộ nhớ cho kênh DMA.

Sau khi cấu hình cho DMA xong, chỉ cần gọi hàm DMA\_Cmd cho ngoại vi tương ứng. Bộ DMA sẽ tự động truyền nhận data cũng như ghi dữ liệu vào vùng nhớ cụ thể.