**CLOCK CONTROL**

**1.Giới thiệu**

Trong vi điều khiển, nguồn clock như là “trái tim” của toàn bộ hệ thống, vì nó cung cấp xung nhịp cho vi điều khiển và các thiết bị ngoại vi giúp chúng hoạt động được. Vì vậy, khi mới bắt đầu tìm hiểu, nghiên cứu bất kỳ một dòng vi điều khiển nào các bạn cần trả lời được 3 câu hỏi:

Có bao nhiêu nguồn Clock cung cấp cho hệ thống?

Đặc điểm của các nguồn Clock?

Cách cấu hình chọn nguồn Clock cung cấp cho hệ thống?

Trong bài viết này, mình sẽ nói giới hạn nguồn Clock của dòng vi điều khiển STM8S.

Cũng như các mạch tích hợp đồng bộ khác, STM8S cũng yêu cầu nguồn cấp xung đồng hồ để hoạt động. Nguồn cấp xung đồng hồ trên dòng vi điều khiển này có thể xuất phát từ 3 nguồn khác nhau:

Nguồn dao động nội LSI (tốc độ thấp – 128kHz).

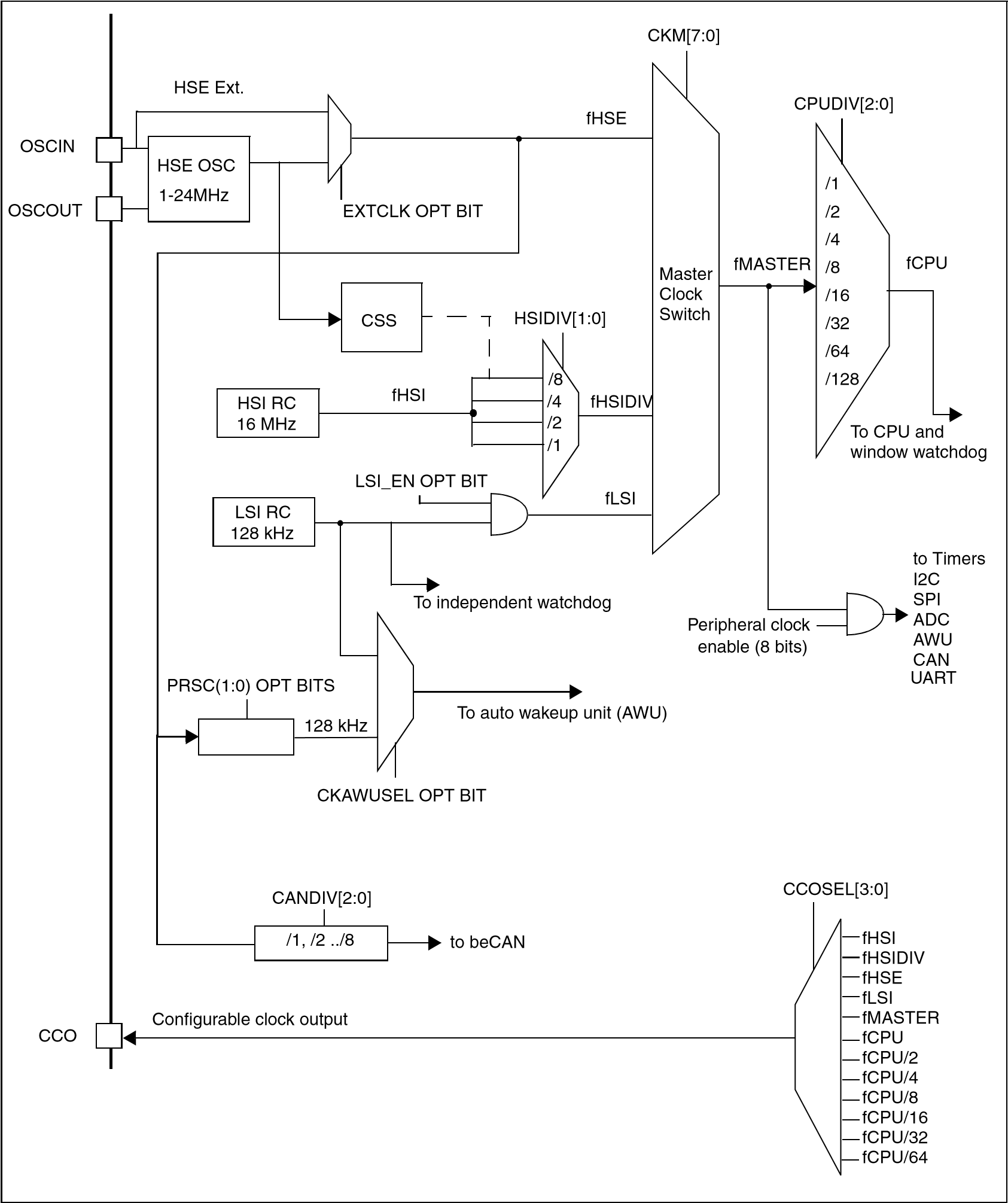
Nguồn dao động nội HSI (tốc độ cao – 16MHz).

Nguồn dao động bên ngoài tốc độ cao HSE (giá trị dao động từ 1 – 16MHz).

Nhìn chung thì các nguồn dao động nội trên STM8S có thể đáp ứng tương đối tốt những công việc cơ bản. Tuy nhiên, đây đều là các mạch dao động RC, về lâu về dài, độ chính xác trong hoạt động không thể đảm bảo. Nên với nguồn cấp lấy từ thạch anh ngoại (HSE) hoặc các thiết bị tạo dao động sẽ là lựa chọn tin cậy hơn.

Lưu ý: Khi mới cấp nguồn hay reset lại thì nguồn Clock của STM8S cấu hình là dao động nội HIS và bộ chia là HSI/8.

Sơ đồ khối Clock STM8S:



**2.Các thanh ghi trong clock**

Ở đây mình chỉ giới thiệu một số thanh ghi quan trọng. Bên dưới có một số thanh ghi mình chưa dịch là do mình chưa hiểu tính năng của bit trong thanh ghi đó. Mong các bạn thông cảm.

**Internal clock register (CLK\_ICKR):** Thanh ghi nguồn clock nội

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | REGAH | LSIRDY | LISEN | FHW | HSIRDY | HSIEN |
|  | | rw | r | rw | rw | r | rw |

Bit 5 **REGAH**: Regulator power off in active halt mode

This bit is set and cleared by software. When it is set, the main voltage regulator is powered off as soon as the MCU enters active halt mode, so the wakeup time is longer.

0: MVR regulator ON in active halt mode

1: MVR regulator OFF in active halt mode

Bit 4 **LSIRDY**: Low speed internal oscillator ready

This bit is set and cleared by hardware. (Bit này can thiệp bằng phần cứng và không thể set hay clear bit này bằng phần mềm (code)).

0: LSI clock not ready (Nguồn clock LSI chưa sẵn sàng)

1: LSI clock ready (Nguồn clock LSI sẵn sàng)

Bit 3 **LSIEN**: Low speed internal RC oscillator enable

This bit is set and cleared by software. (Giá trị của bit này thay đổi phần mềm (code))

0: Low-speed internal RC off. (Tắt nguồn clock LSI).

1: Low-speed internal RC on. (Bật nguồn clock LSI).

Bit 2 **FHWU**: Fast wakeup from halt/active halt modes

This bit is set and cleared by software.

0: Fast wakeup from halt/active halt modes disabled

1: Fast wakeup from halt/active halt modes enabled

Bit 1 **HSIRDY**: High speed internal oscillator ready

This bit is set and cleared by hardware.

0: HSI clock not ready. (Nguồn clock HSI chưa sẵn sàng).

1: HSI clock ready. (Nguồn clock HSI sẵn sàng).

Bit 0 **HSIEN**: High speed internal RC oscillator enable

This bit is set and cleared by software.

0: High-speed internal RC off. (Tắt nguồn clock HSI).

1: High-speed internal RC on. (Bật nguồn clock HSI).

**External clock register (CLK\_ECKR)**: Thanh ghi nguồn clock ngoại

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | | | | | | HSERDY | HSEEN |
| r | rw |

Bit 1 **HSERDY**: High speed external crystal oscillator ready

This bit is set and cleared by hardware.

0: HSE clock not ready. (Nguồn clock HSE chưa sẵn sàng).

1: HSE clock ready. (Nguồn clock HSE sẵn sàng).

Bit 0 **HSEEN**: High speed external crystal oscillator enable

This bit is set and cleared by software. It can be used to switch the external crystal oscillator on or off. It is set by hardware in the following cases:

0: HSE clock off. (Tắt nguồn clock HSE).

1: HSE clock on. (Bật nguồn clock HSE).

**Clock master status register (CLK\_CMSR)**: Thanh ghi trạng thái nguồn clock master

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CKM[7:0] | | | | | | | |
| r | r | r | r | r | r | r | r |

Bits 7:0 **CKM[7:0]**: Clock master status bits

0xE1: HSI selected as master clock source. Nguồn clock HSI đã được chọn cho master clock.

0xD2: LSI selected as master clock source. Nguồn clock LSI đã được chọn cho master clock.

0xB4: HSE selected as master clock source. Nguồn clock HSE đã được chọn cho master clock.

Clock master switch register (CLK\_SWR)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWI[7:0] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw |

Bits 7:0 **SWI[7:0]**: Clock master selection bits

0xE1: HSI selected as master clock source. Set nguồn clock HIS cho master clock.

0xD2: LSI selected as master clock source.Set nguồn clock LIS cho master clock. 0xB4: HSE selected as master clock source. Set nguồn clock HSE cho master clock.

**3.Giải thích hàm trong thư viện stm8s\_clk.**

Ở đây mình chỉ giới thiệu 1 số hàm hay dùng.

|  |
| --- |
| *void CLK\_DeInit(void);*  Hàm này reset các giá trị của các thanh ghi clock về giá trị mặc định. |

|  |  |  |
| --- | --- | --- |
| *void CLK\_HSECmd(FunctionalState NewState);*  Hàm này dùng để bật hay tắt nguồn clock HSE. Bản chất hàm này là là set bit HSEEN của thanh ghi CLK\_ECKR bằng 0 hay 1. | | |
| Tham số | Đối số | Chú thích |
| *NewState* | ENABLE | bật nguồn clock HSE, bản chất hàm này là set bit HSEEN=1 |
| DISABLE | tắt nguồn clock HSE, bản chất hàm này là set bit HSEEN=0 |

|  |  |  |
| --- | --- | --- |
| *void CLK\_HSICmd(FunctionalState NewState);*  Hàm này dùng để bật hay tắt nguồn clock HSI. Bản chất hàm này là là set bit HSIEN của thanh ghi CLK\_ICKR bằng 0 hay 1. | | |
| Tham số | Đối số | Chú thích |
| *NewState* | ENABLE | bật nguồn clock HSI |
| DISABLE | tắt nguồn clock HSI |

|  |  |  |
| --- | --- | --- |
| *void CLK\_LSICmd(FunctionalState NewState);*  Hàm này dùng để bật hay tắt nguồn clock LSI. Bản chất hàm này là là set bit LSIEN của thanh ghi CLK\_ICKR bằng 0 hay 1. | | |
| Tham số | Đối số | Chú thích |
| *NewState* | ENABLE | bật nguồn clock LSI |
| DISABLE | tắt nguồn clock LSI |

|  |  |  |
| --- | --- | --- |
| *ErrorStatus CLK\_ClockSwitchConfig(CLK\_SwitchMode\_TypeDef CLK\_SwitchMode, CLK\_Source\_TypeDef CLK\_NewClock, FunctionalState ITState, CLK\_CurrentClockState\_TypeDef CLK\_CurrentClockState);*  Hàm này dùng để chuyển đổi nguồn clock qua lại giữa LSI, HIS, HSE. | | |
| Tham số | Đối số | Mô tả |
| *CLK\_SwitchMode*  Chế độ đổi clock | CLK\_SWITCHMODE\_MANUAL | Chế độ Manual |
| CLK\_SWITCHMODE\_AUTO | Chế độ Auto |
| *CLK\_NewClock*  chọn nguồn clock bạn muốn sử dụng | CLK\_SOURCE\_HSI | Chọn clock nội HSI |
| CLK\_SOURCE\_LSI | Chọn clock nội LSI |
| CLK\_SOURCE\_HSE | Chọn clock ngoại HSE |
| *ITState* bật hay tắt chế độ ngắt (interrupt) khi chuyển đổi nguồn clock | ENABLE | Cho phép ngắt |
| DISABLE | Không cho phép ngắt |
| *CLK\_CurrentClockState*  đối số này cho phép bạn chuyển sang nguồn clock mới hay vẫn hoạt động ở nguồn lock cũ | CLK\_CURRENTCLOCKSTATE\_DISABLE | Dùng clock cũ |
| CLK\_CURRENTCLOCKSTATE\_ENABLE | Dùng clock mới |

|  |  |  |
| --- | --- | --- |
| *void CLK\_SYSCLKConfig(CLK\_Prescaler\_TypeDef CLK\_Prescaler);*  Hàm này cấu hình bộ chia của clock HSI và CPU. | | |
| *CLK\_Prescaler* | CLK\_PRESCALER\_HSIDIV1 | cấu hình bộ chia của nguồn clock HIS |
| CLK\_PRESCALER\_HSIDIV2 |
| CLK\_PRESCALER\_HSIDIV4 |
| CLK\_PRESCALER\_HSIDIV8 |
| CLK\_PRESCALER\_CPUDIV1 | cấu hình bộ chia của nguồn clock CPU |
| CLK\_PRESCALER\_CPUDIV2 |
| CLK\_PRESCALER\_CPUDIV4 |
| CLK\_PRESCALER\_CPUDIV8 |
| CLK\_PRESCALER\_CPUDIV16 |
| CLK\_PRESCALER\_CPUDIV32 |
| CLK\_PRESCALER\_CPUDIV64 |
| CLK\_PRESCALER\_CPUDIV128 |

Ví dụ:

CLK\_SYSCLKConfig(CLK\_PRESCALER\_HSIDIV1);// cấu hình bộ chia của HIS là 1

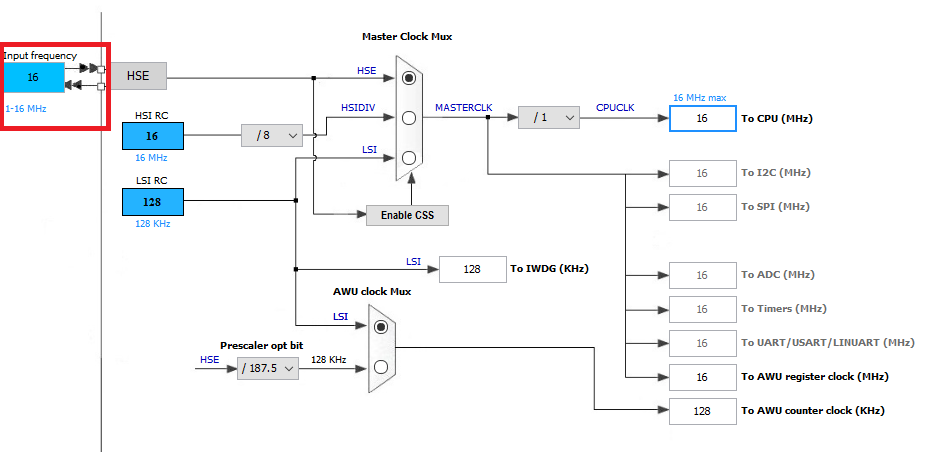
CLK\_SYSCLKConfig(CLK\_PRESCALER\_CPUDIV2);// cấu hình bộ chia của CPU là 2

|  |
| --- |
| *uint32\_t CLK\_GetClockFreq(void);*  Hàm này trả về tần số hoạt động của nguồn clock đang sử dụng. |

**4.Cấu hình nguồn lock**

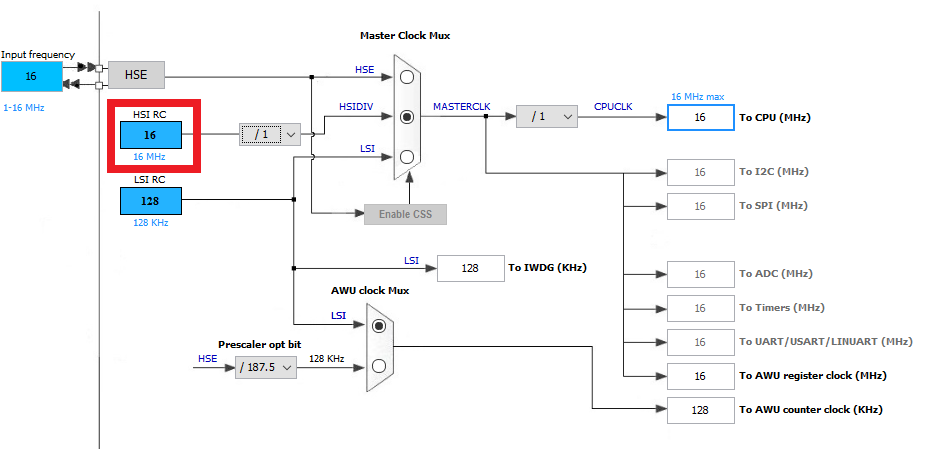
a.Cấu hình nguồn clock HSE

Trong trường hợp giải sử bạn gắn thạch anh ngoại là 16MHz vào STM8S103 thông qua 2 chân: PA1(OSCIN) và PA2(OSCOUT).



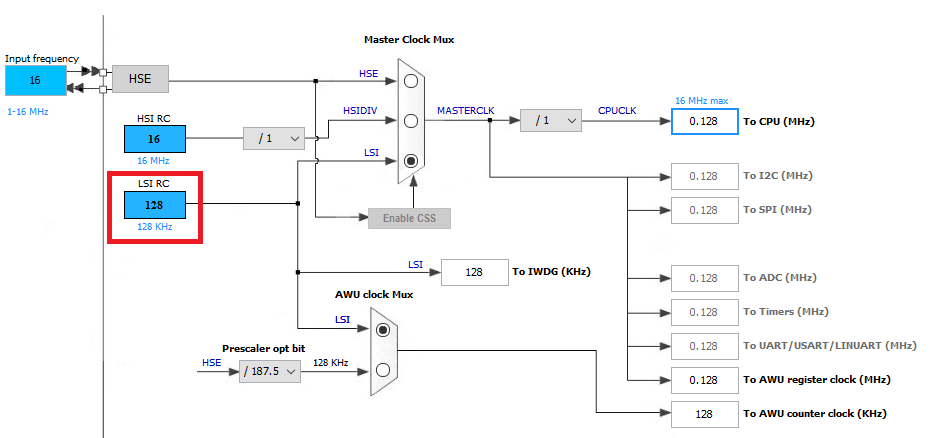
|  |
| --- |
| void CLK\_HSE\_16MHz\_Init(void)  {  /\* reset lại tất cả các thanh ghi của Clock \*/  CLK\_DeInit();  /\* cấu hình Fcpu ở chế độ chia 1, thì Fcpu vẫn là 16MHz \*/  CLK\_SYSCLKConfig(CLK\_PRESCALER\_CPUDIV1);  /\* chuyển nguồn clock qua dùng HSE \*/  CLK\_ClockSwitchConfig(CLK\_SWITCHMODE\_AUTO,CLK\_SOURCE\_HSE, DISABLE, CLK\_CURRENTCLOCKSTATE\_DISABLE);  } |

b.Cấu hình nguồn clock HIS



|  |
| --- |
| void CLK\_HSI\_16MHz\_Init(void)  {  /\* reset lại tất cả các thanh ghi của Clock \*/  CLK\_DeInit();  /\* cấu hình Fhsi ở chế độ chia 1, thì Fmaster là 16MHz \*/  CLK\_SYSCLKConfig(CLK\_PRESCALER\_CPUDIV1);  /\* cấu hình Fcpu ở chế độ chia 1, thì Fcpu vẫn là 16MHz \*/  CLK\_SYSCLKConfig(CLK\_PRESCALER\_HSIDIV1);  /\* chuyển nguồn clock qua dùng HSI\*/  CLK\_ClockSwitchConfig(CLK\_SWITCHMODE\_AUTO,CLK\_SOURCE\_HSI, DISABLE, CLK\_CURRENTCLOCKSTATE\_DISABLE);  } |

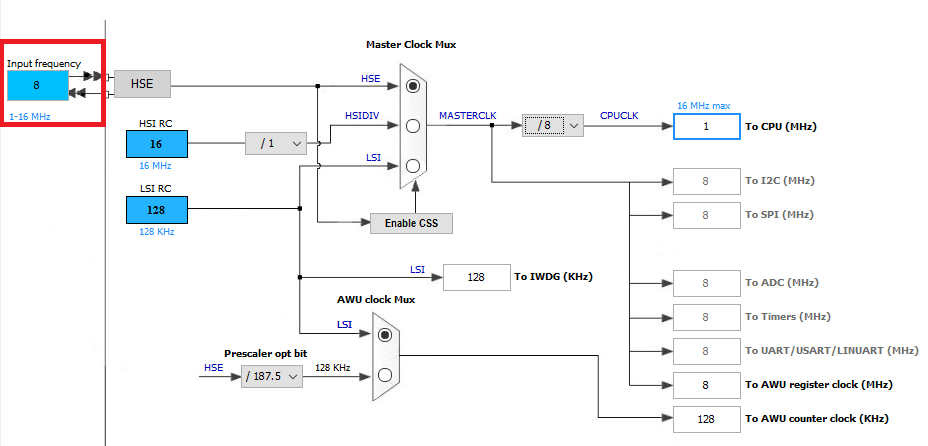
c.Cấu hình nguồn clock LSI



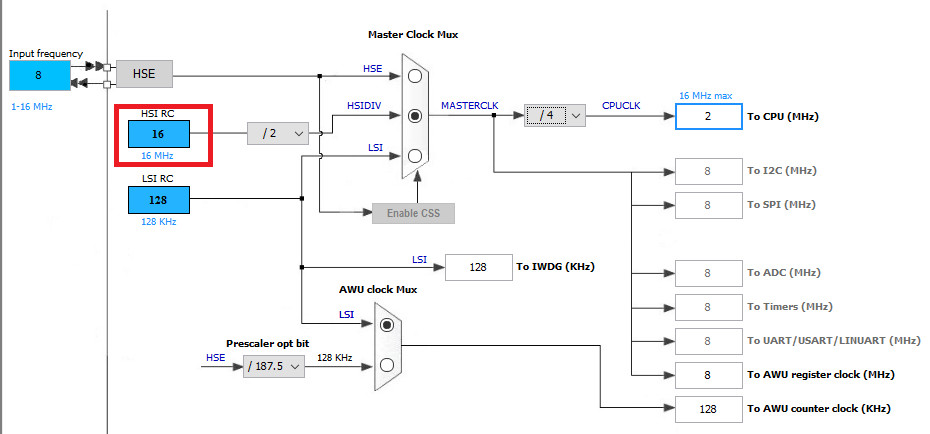
|  |
| --- |
| void CLK\_LSI\_128kHz\_Init(void)  {  /\* reset lại tất cả các thanh ghi của Clock \*/  CLK\_DeInit();  /\* cấu hình Fcpu ở chế độ chia 1, thì Fcpu vẫn là 128kHz \*/  CLK\_SYSCLKConfig(CLK\_PRESCALER\_CPUDIV1);  /\* chuyển nguồn clock qua dùng LSI\*/  CLK\_ClockSwitchConfig(CLK\_SWITCHMODE\_AUTO,CLK\_SOURCE\_LSI, DISABLE, CLK\_CURRENTCLOCKSTATE\_DISABLE);  } |

5.Bài tập tự luyện:

1.Sử dụng nguồn thạch anh ngoài 8MHz và cấu hình Fcpu ở chế độ chia 8.



2.Cấu hình thạch anh nội HSI ở chế độ HIS chia 2 và Fcpu chia 4.



3.Cấu hình thạch anh nội LSI ở chế độ Fcpu chia 2.

