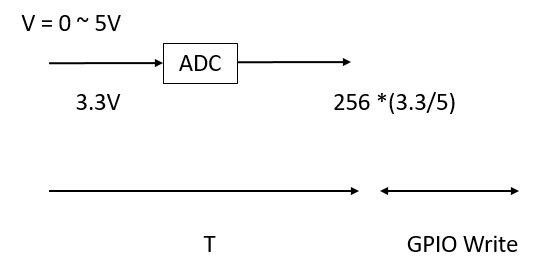
ADC là ngoại vi invert tín hiệu điện áp bên ngoài thành tín hiệu số

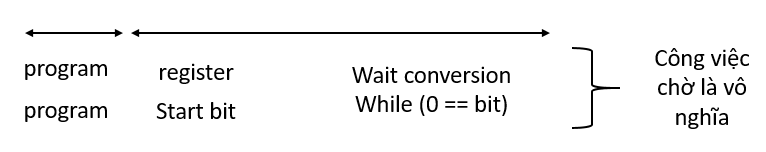


Int ADC\_Conversion()

{

Return kết quả đo được (thời gian thực hiện lớn)

}

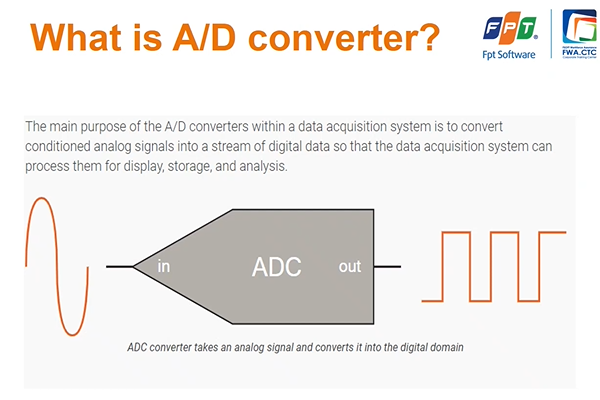


Dùng GPIO write thì khi nào thực sự ghi lên chân đó

ADC

Hiểu cơ bản về bộ biến ADC

ADC trên 1 module KL46 và cách tạo 1 project đơn giản về ADC

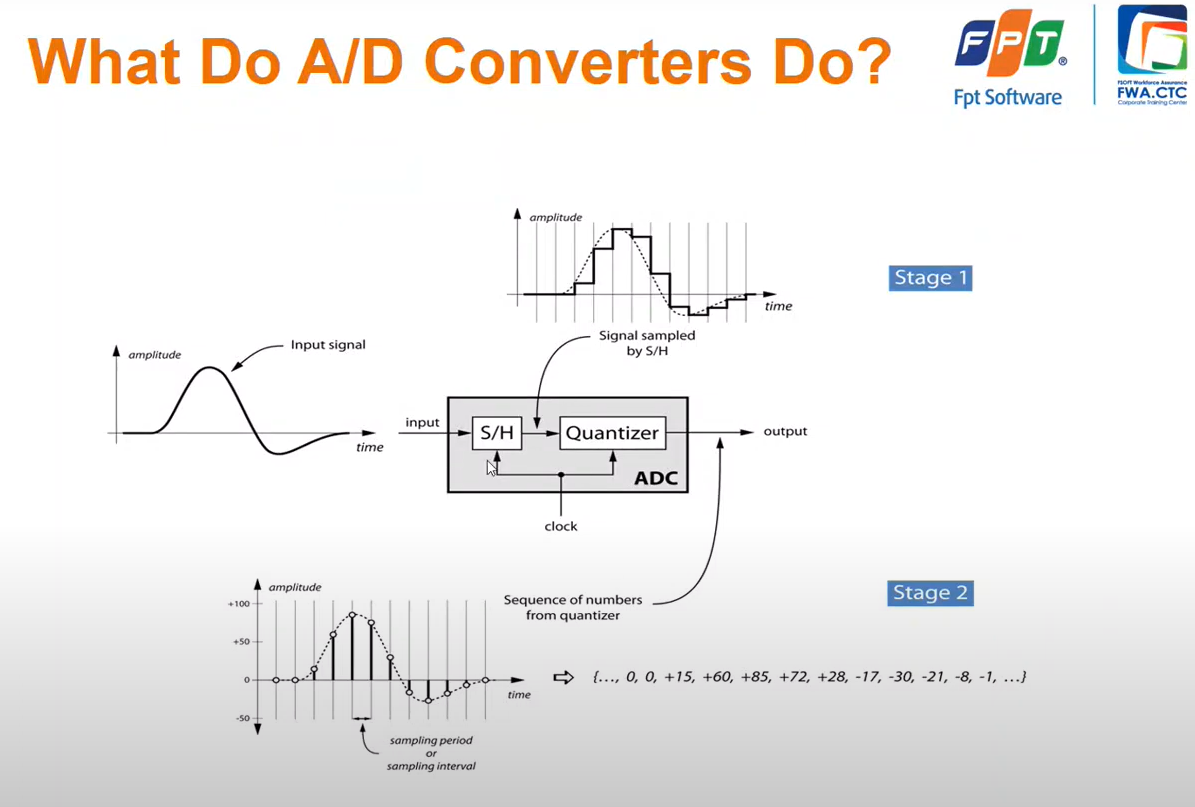


Trên thực tế, các giác quan trên cơ thể cta sẽ cảm nhận độ thay đổi của nhiệt độ, độ sáng tối của ánh sáng, mùi vị. Tất cả đều là giá trị tương tự. làm sao để chuyển đổi các giá trị tương tự này thành giá trị số thì trên vdk dùng bộ biến đổi ADC này.

Đo nhiệt độ, độ ẩm, đọc qua module cảm biến gia tốc.

Đọc mic

Đặc điểm chung của các ứng dụng vừa nêu, đầu vào là 1 giá trị điện áp, đầu ra là 1 giá trị số, input của nó là 1 tín hiệu tương tự, output của nó là 1 giá trị số

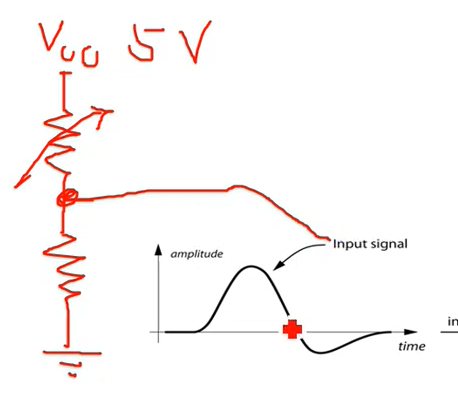


Trong bộ ADC này có những gì và vì sao chuyển đổi được từ giá trị tương tự sang giá trị số.

Tín hiệu tương tự/ đầu vào input cho bộ ADC của mình. Khối S/H là khối lấy mẫu, còn khối Quantizer là lượng tử hoá và cả hai cần clock đầu vào để hoạt động.

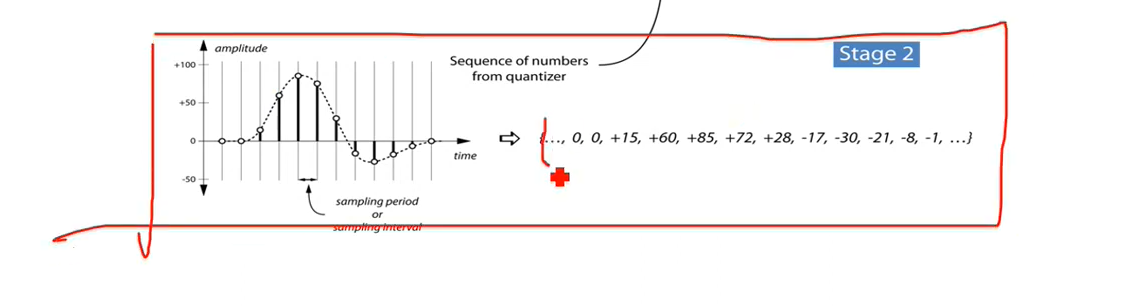
Từ tín hiệu tương tự, nó sẽ được lấy mẫu, thời gian giữa 2 lần lấy mẫu này gọi là chu kỳ lấy mẫu. 1 bên chu kỳ lấy mẫu được gọi là tần số lấy mẫu.

Ví dụ mình có 1 con quang trở và 1 trở cố định, GND, Vcc 5V.



Điểm mình lấy que đo đưa vào là input signal này thì nó sẽ đo độ thay đổi của ánh sáng là bao nhiêu, ra ngoài trởi nắng thì 80K nấc, tối thì 0 nấc, 5 – 10 – 50 – 100 nấc, ánh sáng đèn là 200 – 300 nấc thì khi có ánh sáng mặt trởi chiếu vào quang trở thì điện trở của nó trong này sẽ giảm. khi mình che tay lại thì điện trở tăng lên tối đa tuỳ vào quang trở.

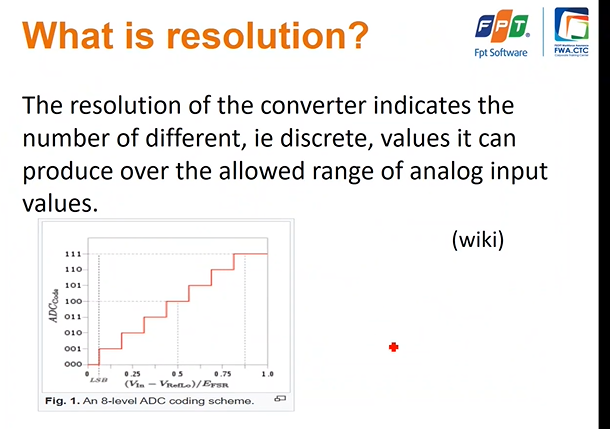
Khi điện trở giảm xuống/ mặt trởi chiếu vào thì lúc đó điện áp chia ra ở trên 2 con trở tương đương trên này 1 con trở khoảng giá trị 100 – 200 ôm tuỳ vào lượng ánh sáng chiếu vào. ở dưới cho là 1K thì mình chia áp ra theo mạch cầu phân áp thì điện áp này sẽ cao. Khi ánh sáng chiếu vào yếu thì điện áp giảm xuống. đó là đầu vào tương tự khi mình đo âm thanh và khối S/H sẽ lấy mẫu và chuyển qua xem sẽ lượng tử hoá.



Bên trái là mảng output giá trị số của bộ ADC từ giá trị 0, dần dần tăng lên để biến đổi 1 giá trị ADC thành 1 giá trị số thì cần 1 khoảng thời gian

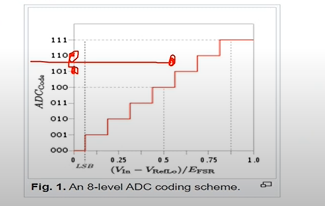
 gọi là chu kỳ lấy mẫu.

Cần tìm hiểu trong bộ ADC này lấy mẫu như thế nào, nó lượng tử ra sao, cách nó hoạt động, thêm những khối nào.



Độ phân giải, ở đây ngta lấy 1 ví dụ bộ ADC có 8 mức

1 giá trị bất kỳ cta lấy mẫu đặt vào thì nó nằm ở mức nào thì out put ra là 1 giá trị số, lân cận theo mức đó, cận trên hay cận dưới thì các slide sau sẽ biết

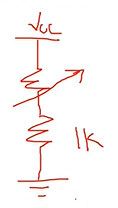


Độ phân giải rất quan trọng trong ví dụ ADC

Cân nặng của cta từ vài chục đến vài trăm kg nhưng khi cân vàng hay rau thì những cái nhẹ hơn chỉ tính tới lạng thì lúc đó độ phân giải càng nhỏ càng chính xác.

1 bộ ADC 8 bit thì cho ta 2^8 mức, ADC 10 bit thì cho 2^10 mức

12 bit thì cho 4096 mức, mức này càng nhỏ thì độ phân giải càng nhỏ càng chính xác hơn.

với ADC 8 bit, từ GND đến Vcc là 0 - 5V, nó sẽ chi cho chúng ta 2^8 mức, 1 bước nhảy của tín hiệu là 5V/256 thì mới detect, phát hiện ra được và bước nhảy giữa giá trị trước và sau rất lớn nhưng mà cần bộ chính xác cao hơn thì cần ADC có số bit cao hơn như là 12 bit hay là 16 bit.

12 bit cho 4096 mức

Độ phân giải càng nhỏ càng chính xác.

Quang trở, con trở 1 K ôm

Khi xài bộ adc 8 bit

Từ vcc đến GND (0 tới 5V)

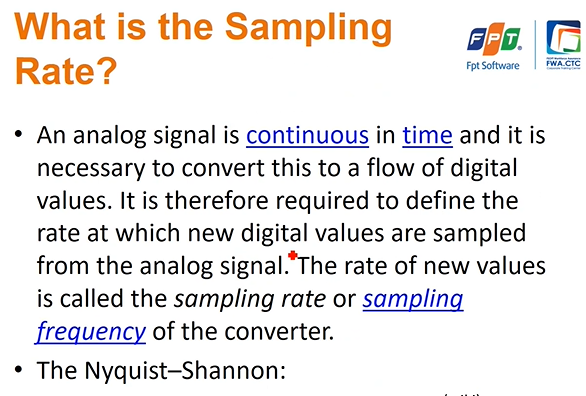
5/256 mức

1 bước nhảy là 5V/256 thì mới detect, phát hiện ra được

Bước nhảy giữa giá trị trước và sau là lớn

Càng ADC nhiều bit càng chính xác.

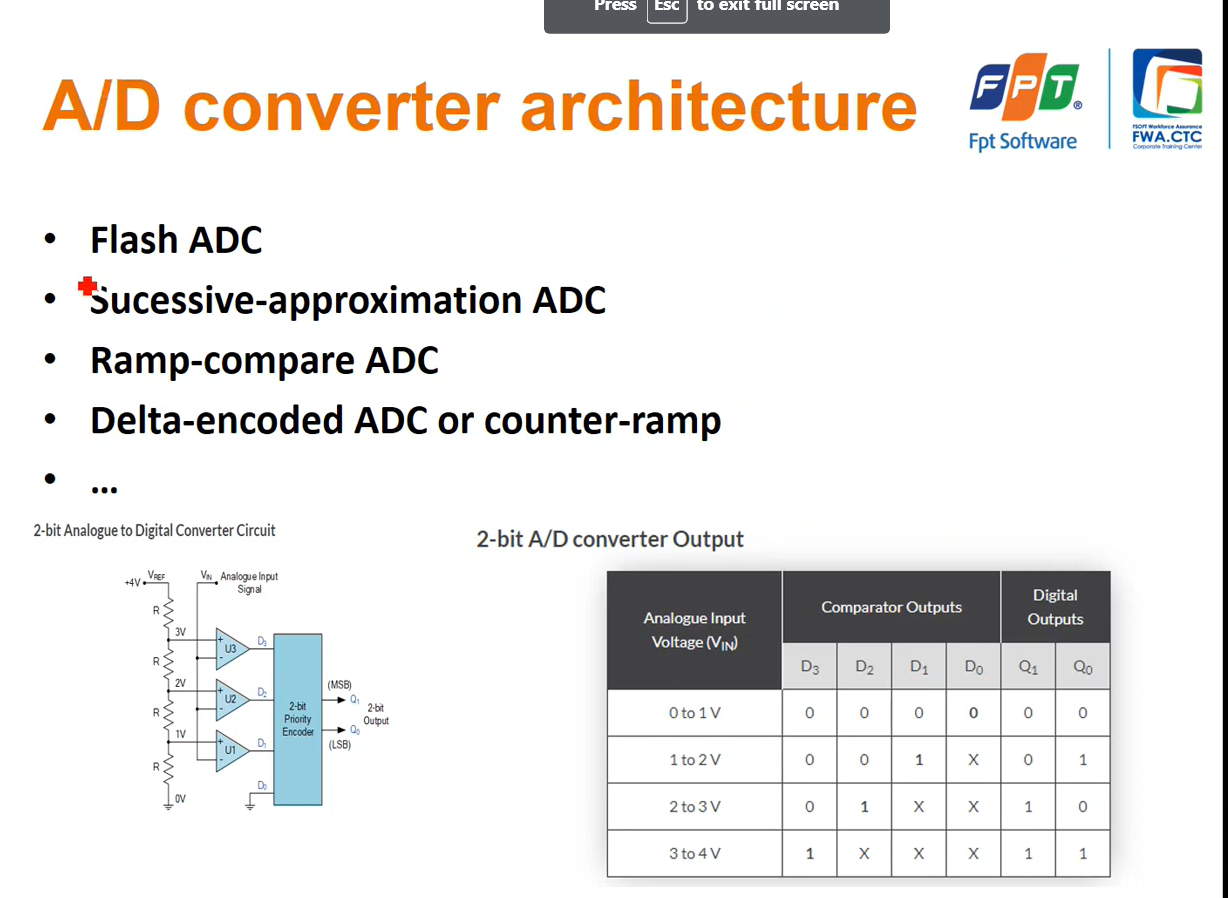
Tốc độ lấy mẫu



Nyquist – Shannon search thêm

Định lý: 1 tín hiệu analog có tần số giới hạn đã được lấy mẫu thì nó có thể được tái tạo hoàn toàn từ 1 chuỗi vô số các mẫu nếu tỷ lệ lấy mẫu lớn hơn 2 lần tần số lấy mẫu.

Trọng tâm: tìm hiểu các kiến trúc trong bộ ADC.



Kiến trúc nào ưu điểm và nhược điểm hơn, ưu điểm nhược điểm của từng cái. So sánh và con KL46 dùng kiến trúc nào?

Ở đây lấy 1 bộ adc 2 bit, ở đầu vào Vin này, ở đây có các opamp hình tam giác.

Khi điện áp chân + lớn hơn – thì output ở mức cao và ngược lại

Opamp có nhiều chế độ nhưng ở đây xài chế độ so sánh. Tức là áp chân cộng lớn hơn chân trừ thì out mức high và ngược lại, còn điện áp ref 4V mà qua mỗi mạch cầu phân áp thì tại mỗi điểm này chênh nhau 1V, 0,1,2,34. Đưa điện áp ref vào từng opamp

Điện áp chân trừ nhỏ hơn chân cộng thì output ra là 0000

Compare output của 3 opamp đều 0 thì digital output ra là 00.

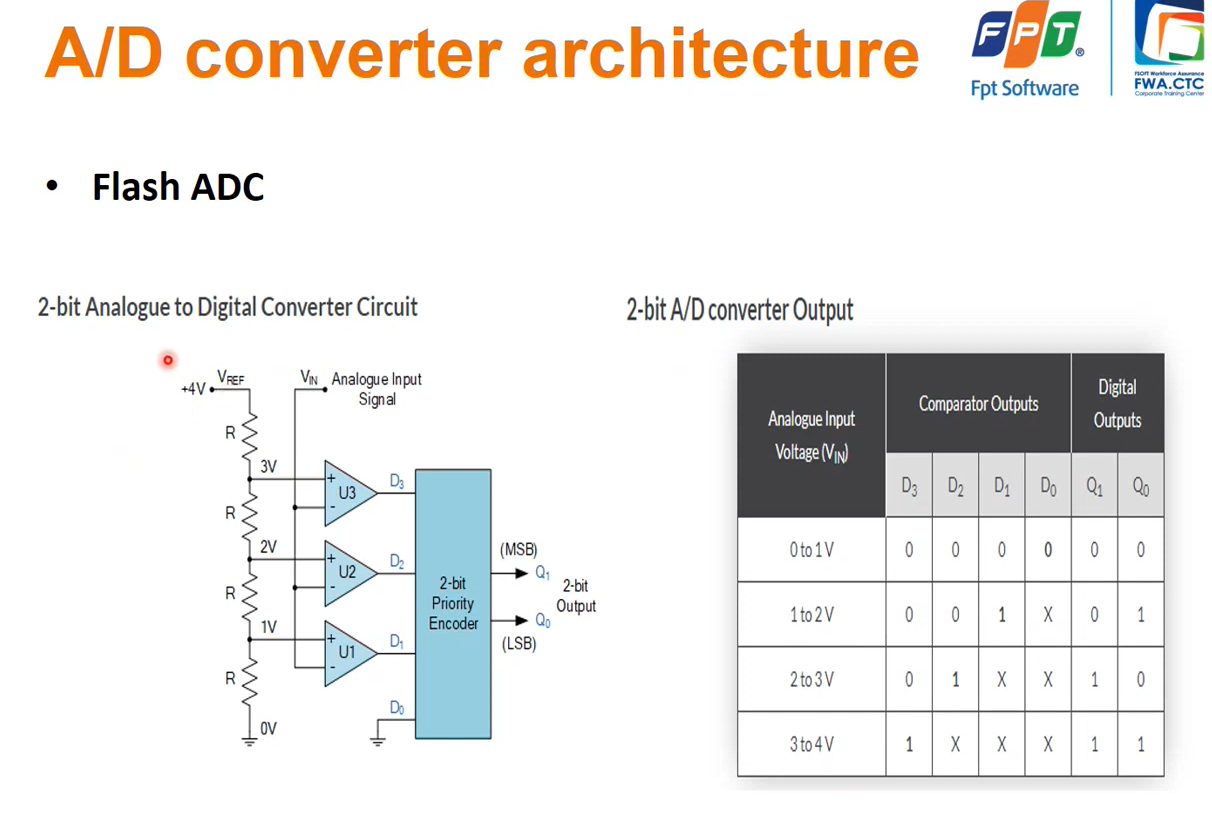
Khi điện áp đưa vào Vin giả sử là 1 phẩy mấy volt thì điện áp chân trừ 1 phẩy Mấy lớn hơn 1.

Lúc đó điện áp chân trừ lớn hơn điện áp chân cộng thì lúc này D1 trả ra giá trị 0

Giá trị các chân D3, D2 này trả ra là 1 thì compare output D3 D2, sẽ là 1 1 00

Cái này hơi ngược (do bảng sai, bỏ vào cho tượng trưng)

Tương tự như lúc nãy so sánh



Nếu Vin khoảng 3,5 Volt thì sau bộ biến đổi ADC này Q0 Q1 là 00 vì điện áp các chân trừ đều lớn hơn ở chân cộng (3, 2, 1V) nên là output ra của các con opamp này đều là 000

Và D0 là sẵn rồi

0000 -> 00

Đó là Flash ADC

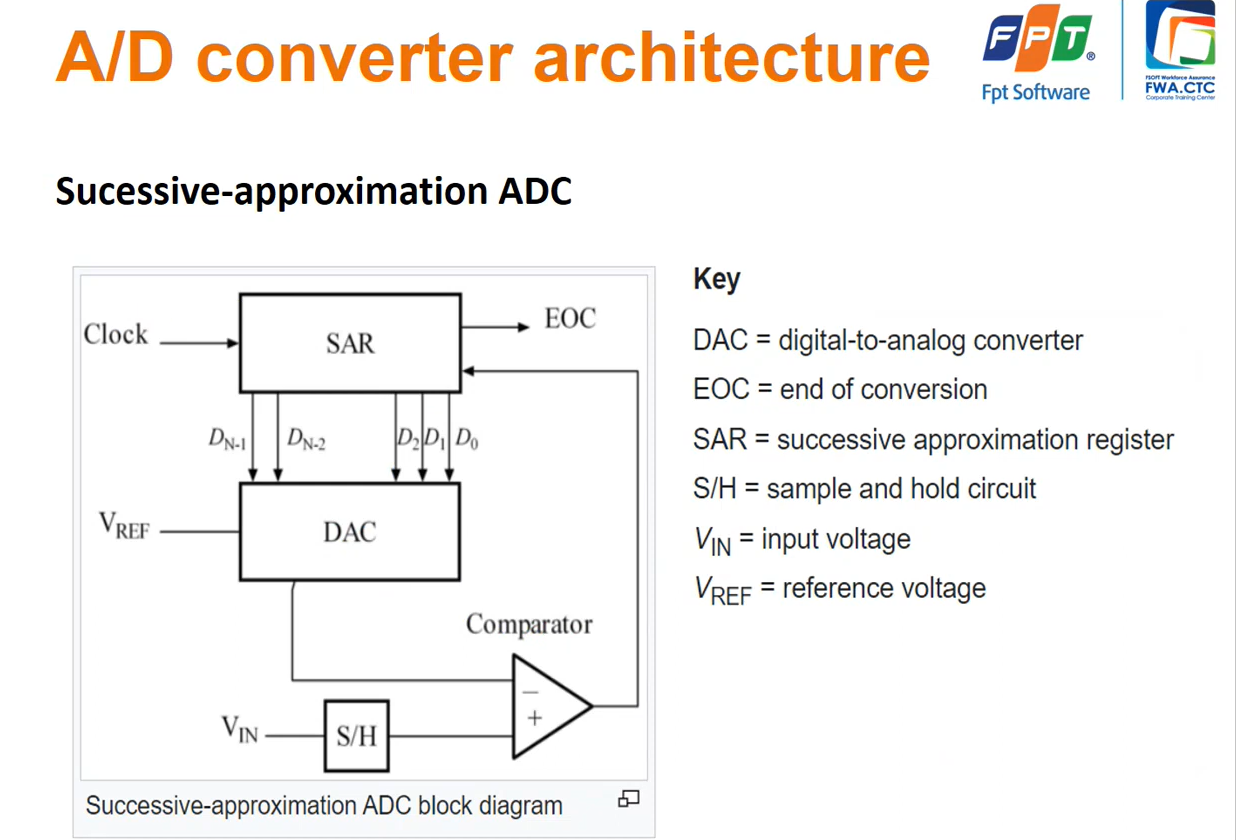
Ko nhìn theo bảng vì bảng đó có thể sai

Điện áp các chân – đều lớn hơn các chân +

Nhận xét:

Bộ biến đổi này có tốc độ nhanh, mình cứ đưa điện áp từ chỗ Vin này thì các con này đều hoạt động, so sánh đồng thời và output đưa ra thì cùng 1 lúc.

Bộ adc nhận giá trị cùng lúc, ko mất quá nhiều thời gian để biến đổi



Bộ biến đổi ADC xấp xỉ

Khối DAC ngược lại với ADC chuyển đổi giá trị số thành giá trị analog thì ngta nhận thấy bộ DAC này thực hiện biến đổi từ số sang analog nhanh hơn rất nhiều so với bộ ADC cho nên ngta đưa bộ DAC này vào trong bộ ADC để nó giúp chuyển đổi giá trị ADC được nhanh chóng, nó nhanh như thế nào thì cta sẽ cùng phân tích.

Điện áp Vin được khối S/H lấy mẫu (Symbol and hold) đưa vào để so sánh.

Lúc này bộ biến đổi xấp xỉ, bộ SAR, đầu vào clock các giá trị từ d0 đến dn-1 tăng dần lên. Đầu tiên là nó từ 000 hết, sau đó sẽ tăng dần lên 1 2 3 cho tới hết giá trị của nó.

Đó là đầu vào của DAC. ở đây các bus này sẽ là đầu vào bộ DAC này. DAC sẽ nhận các giá trị số từ bộ SAR đưa qua, thì nó sẽ out ra giá trị analog (vì nó biến đổi từ số qua analog) thì output đó được đưa vào để so sánh

Output của opamp được đưa vào SAR này, EOC (end of conversion), là cờ để báo quá trình chuyển đổi kết thúc.

Khi các giá trị D0 D1 D2 cho đến Dn-1 tăng dần lên thì output DAC tăng dần lên cho tới khi nó lớn áp của Vin (lớn hơn áp mà chúng ta đã lấy mẫu) thì output của opamp này sẽ thay đổi.

Ban đầu tăng từ 0 lên, giả sử như cái Vin này ban đầu có 1 giá trị nào đó nó đang lấy lớn hơn cái output của DAC. Ban đầu bên SAR này đếm dần lên, đếm dần lên thì out put tăng từ 0 dần lên thì điện áp chân “trừ” đang thấp hơn điện áp chân “Vin” thì con opamp này, đầu ra opamp đang ở mức cao.

Điện áp chân cộng đang lớn hơn thì đầu ra opamp ở mức cao thì SAR đếm dần lên thì output DAC tăng dần lên cho đến khi lớn hơn điện áp “Vin” này, nghĩa là điện áp chân – vừa lớn hơn điện áp chân + thì lúc đó cái output chuyển từ mức cao xuống thấp thì các bộ SAR này lưu giá trị (D0 D1 D2…Dn-1) và xem các giá trị này là output của nó và nó báo là đã chuyển đổi xong

Out put mà mình nhận được là bộ các giá trị D0 D1 D2…Dn-1 này.

Đây là ý tưởng của bộ ADC xấp xỉ.

1 bước tăng bao nhiêu tuỳ thuộc DAC. Bộ DAC, 1 bước tăng càng nhỏ (độ phân giải) thì kết quả nhận được càng chính xác. Phụ thuộc vào bộ DAC này.

Bộ này có nhược điểm so với bộ flash. Bộ Flash thì vào 1 cái là có luôn (ra được output luôn), ko mất thời gian như bộ ADC xấp xỉ

ADC xấp xỉ tốn thời gian tăng từ 0 đến 1 giá trị xấp xỉ nào đó cho đến khi output của DAC bằng với Vin, nó vừa vượt qua thằng Vin

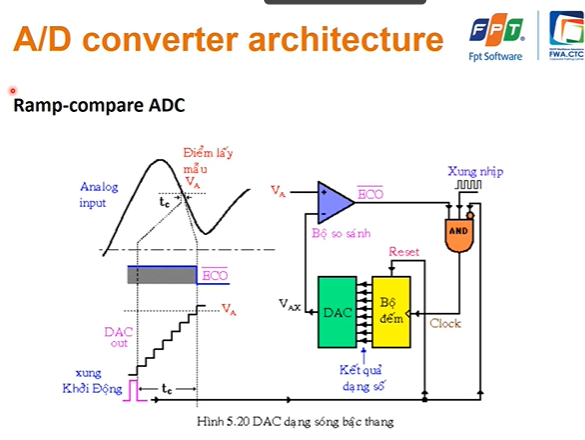
Tốn thời gian ở chỗ đó

Mới vào tăng từ 0 rồi tăng dần, áp tăng dần cho đến khi vừa vượt qua Vin 1 cái thì nó sẽ xem là giá trị bộ SAR này sẽ là output của bộ biến đổi ADC của mình 🡪 mất thời gian ở khúc tăng.

Ví dụ nếu Vin rất lớn thì D0 … Dn-1 tăng mãi đến khi vừa vượt qua Vin.

Nếu Vin rất bé hoặc bằng 0 luôn thì từ 0 tăng lên 1 cái là vừa vượt qua Vin, giá trị đó là output của bộ ADC luôn thì thời gian tăng đó ngắn.

T biến đổi nhanh hay chậm phụ thuộc vào Vin nữa.



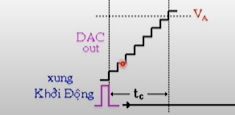
Tương tự bộ vừa rồi

Sơ đồ kiến trúc ramp compare cải thiện được kiến trúc của thằng trên ở chỗ nào.

Bộ đếm ở đây cũng đếm dần lên. Lấy output của bộ đếm làm input cho bộ DAC. Bộ DAC nhận input này và nó out ra giá trị để đưa vào bộ so sánh. Nhưng ở đây có thể 1 cổng AND nữa.

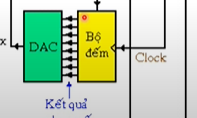
Phân tích từ đầu

Giá trị điện áp tương tự, bộ đếm sẽ đếm thì cái Vax, cái output của DAC tăng dần lên

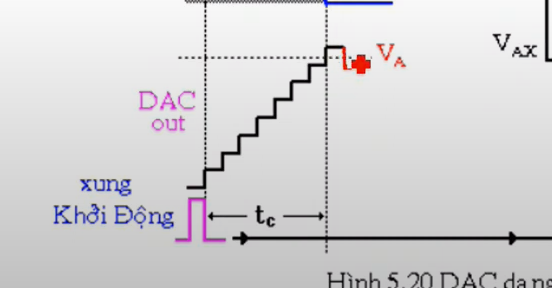
, cứ đếm từ 0 lên, tăng dần lên cho tới 1 lúc vượt quá giá trị VA



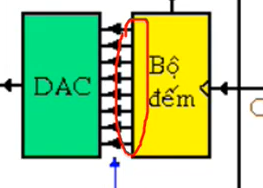
1 khi nó lớn hơn giá trị VA out put BCO đang ở mức cao xuống mức thấp nó vào cổng AND. Nếu đầu vào của nó là 0 thì đầu ra cũng là 0 (0 and với bao nhiêu cũng là 0) thì đầu ra cổng AND là 0, đồng thời BCO cũng là tín hiệu để lấy/ báo cho cta đã biết là kết thúc chuyển đổi (quá trình chuyển đổi kết thúc) vì VAX vừa vượt qua VA thì nó sẽ lấy giá trị out put của

D0 đến Dn-1 là kết quả của biến đổi ADC, sau đó bộ đếm này tiếp tục. với các mẫu sau này mình lấy thì nó ko đếm từ 0 lên nữa mà lúc đó nó sẽ đếm từ điểm mà nó kết thúc trước đó.

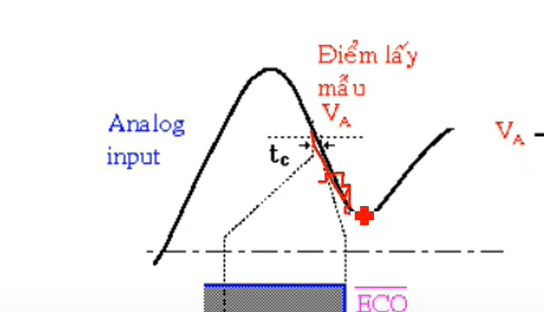
Ngõ ra bộ AND này lúc nãy mình phân tích là nó đang là 0, 0 and bao nhiêu cũng là 0, đưa vào đây thì bộ đếm lại đếm xuống bởi vì VAX này nó đang lớn hơn VA (vừa vượt qua VA) thì nó lại AND cho cta biết là out put của bộ đếm này đếm xuống thì clock này lại đếm xuống.

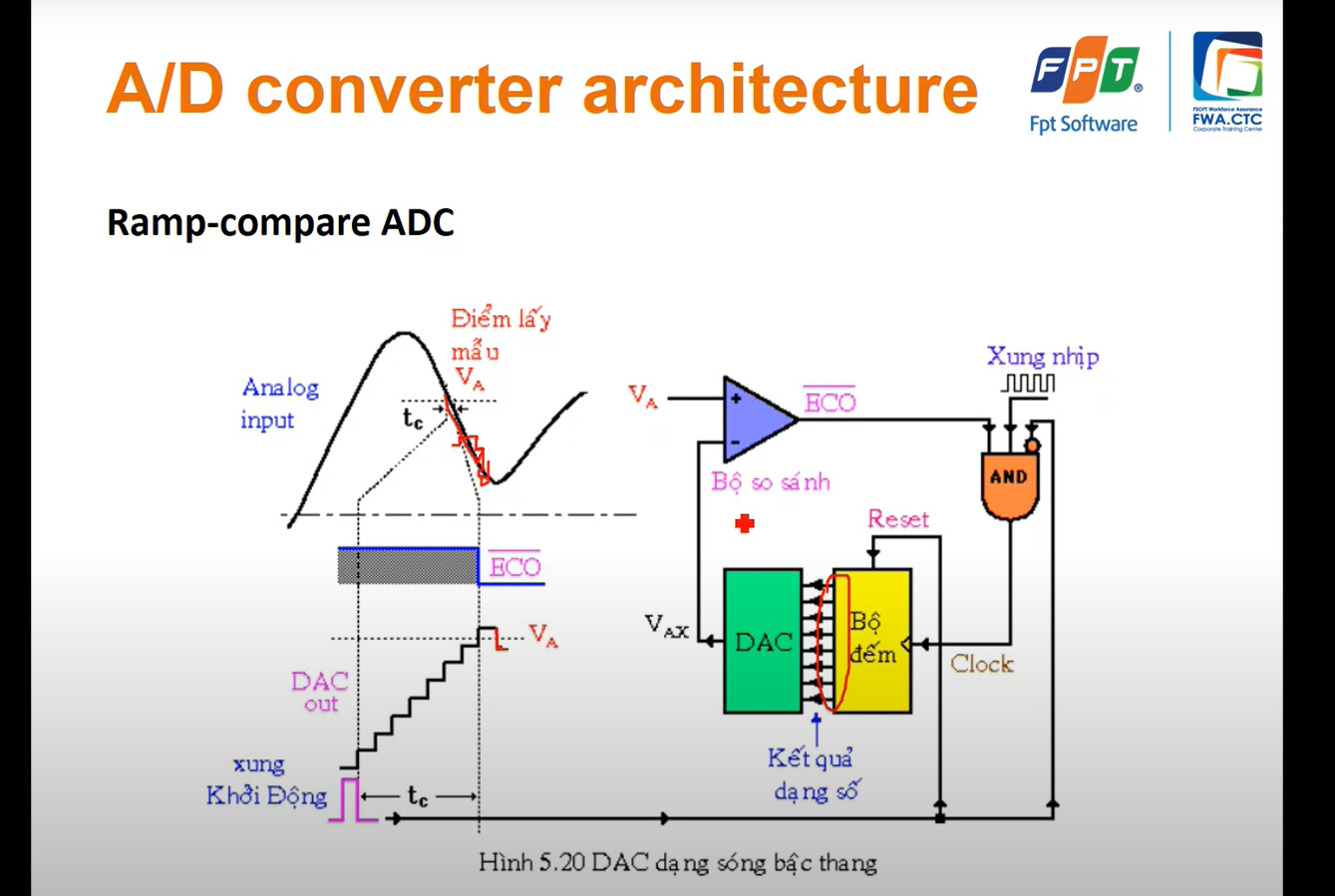


Thì nó lại vừa vượt qua VA thì AND nó báo cho cta biết out put này là giá trị

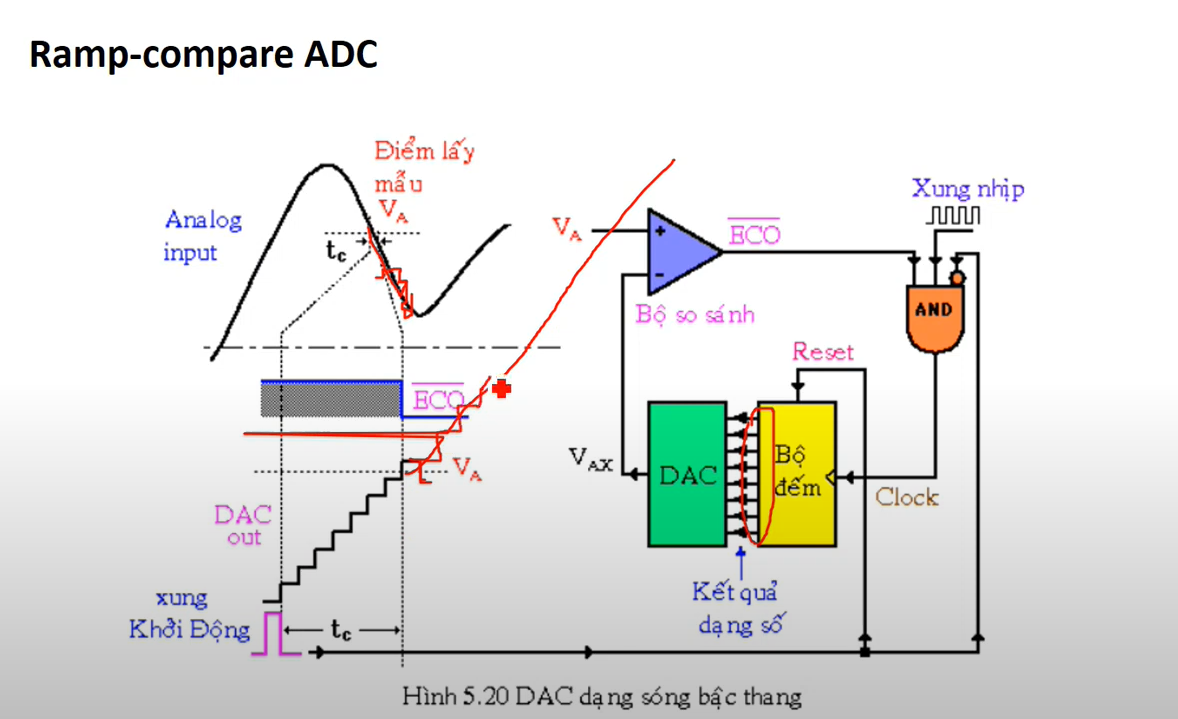
của lần chuyển đổi vừa xong

Thì đồ thị của chúng ta đi xuống thì cái output của cta vừa đếm lên đếm xuống ntn thì nó sẽ bám lấy điểm này

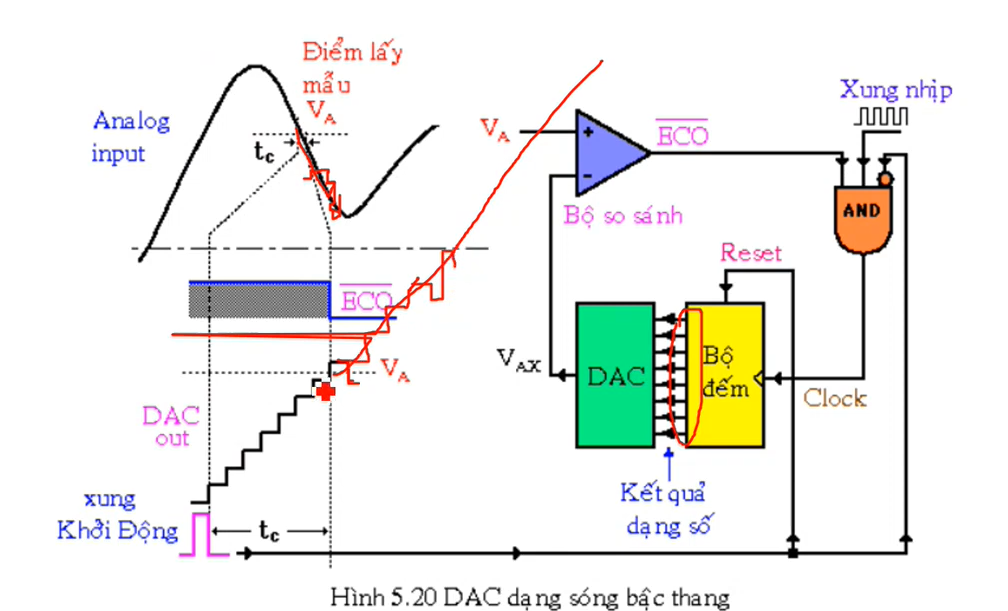
 mãi, nó sẽ bám lấy thì quá trình đổi nhanh hơn, nó sẽ ko đếm từ 0 nữa mà là đếm từ giá trị mà nó kết thúc trước đó nghĩa là điểm nó bắt đầu là điểm lân cận với giá trị output của chúng ta rồi thì nó sẽ nhanh hơn.



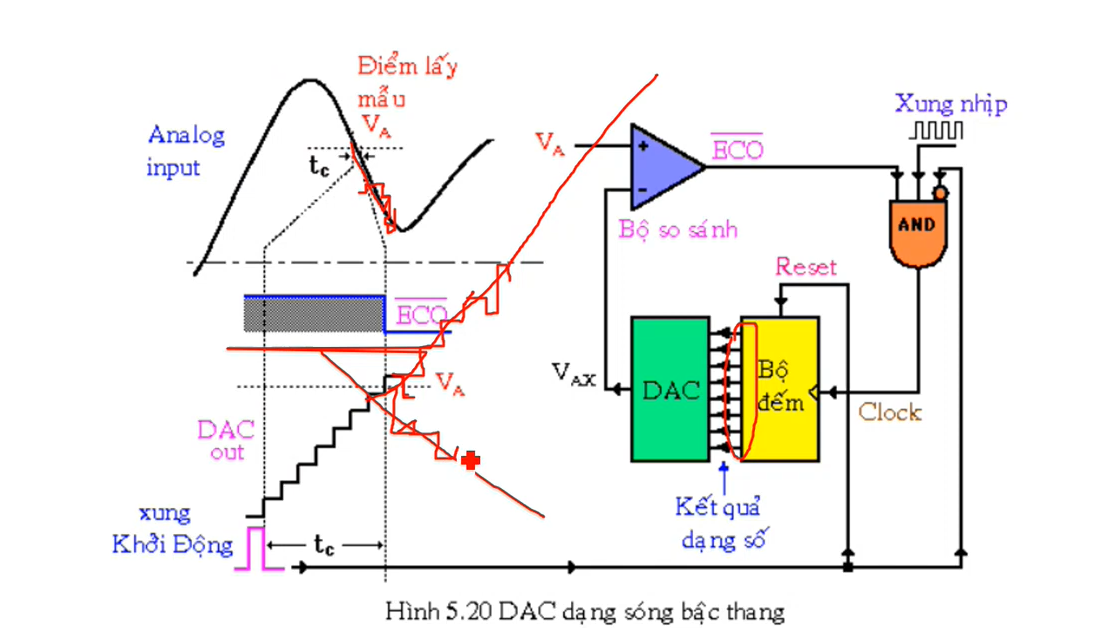
Kiến trúc này cải thiện được điểm của thằng xấp xỉ này. Lần nào cũng đếm từ 0 lên, nó sẽ lâu. Còn thằng này sẽ đếm từ điểm kết thúc trước đó. Nếu như VA đang còn tăng lên



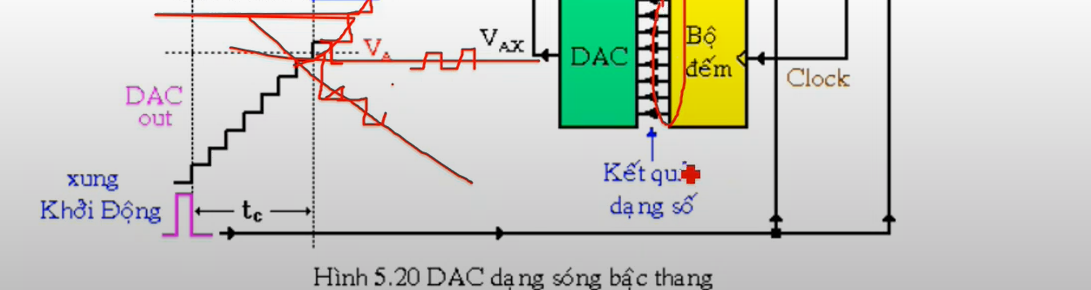
thì nó cứ bắt đến điểm này, vượt qua cái thì lại đếm xuống, nếu thấy thấp hơn thì lại đếm lên



Còn VA cứ đếm xuống thì nó lại bắt lấy đồ thị này thôi.

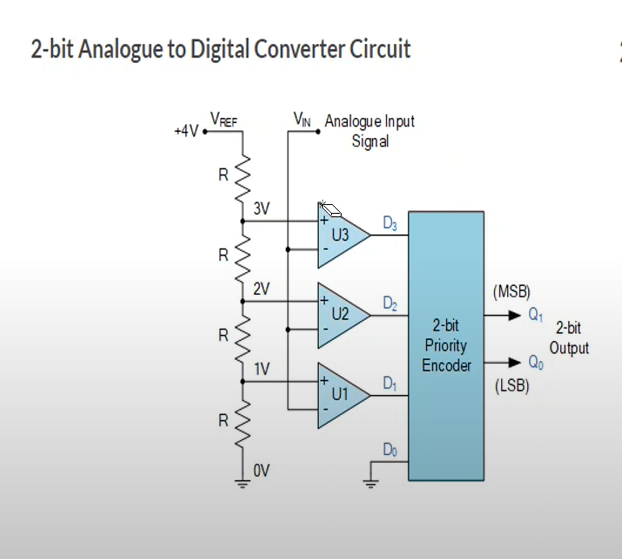


Còn nếu VA đi ngang thì nó cứ bắt lấy như thế này



Cho nên giúp cho tốc độ chuyển đổi rất nhanh.

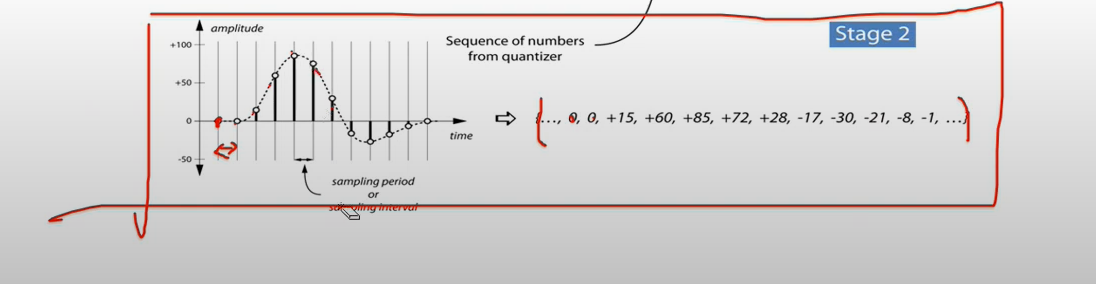
Thằng FLASH cũng nhanh nhưng vì sao ngta ko/ít áp dụng nó. Cái này sẽ làm cho chúng ta tốn tài nguyên, ví dụ như là khi mình làm 1 phép tính

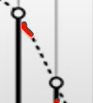


Ví dụ bộ này là 8 bit thì tính xem tốn bao nhiêu Opamp này, bao nhiên điện trở

ở đây D0 nối đất rồi nên là tốn 2^8 – 1 thôi. Điện trở tốn 2^8

nếu ở đây 8 bit chỉ tốn mấy trăm thôi nhưng các bộ ADC trải trên các vdk thường là 10 bit, 12 bit. Đó là trong vdk. Còn những con nằm ngoài như bộ ADS115 hay là các con nằm trong các cân đo thì nó sẽ phải là 16 bit hay 30 bit (rất lớn 🡪 tốn nhiều tài nguyên).

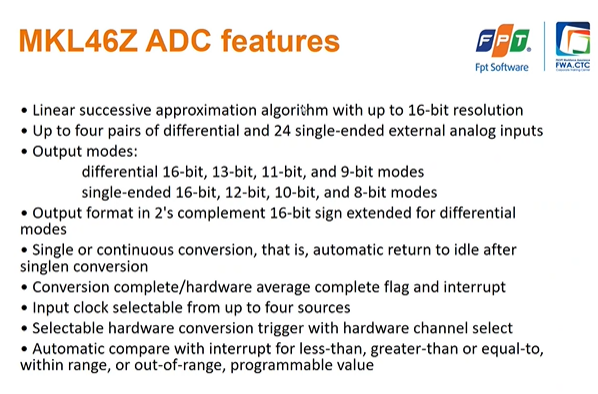
Khoảng thời gian giữa 2 chu kỳ lấy mẫu được gọi là, nếu lấy mẫu được nhanh hơn thì sẽ ko bỏ qua các giá trị ở giữa này



Nghĩa là nếu lấy mẫu chậm thì các bước nhảy này sẽ lớn. nếu lấy mẫu chậm mà có độ phân giải cao thì nó cũng ko giúp chúng ta lấy được mẫu chính xác hơn.

Qua con KL46

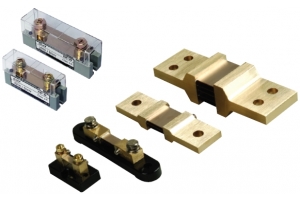
Trên module này



Sử dụng thuật toán xấp xỉ với độ phân giải 16.

Có 4 bộ Differential (so sánh giữa 2 chân). Ví dụ đo dòng điện thì thường cho dòng điện chạy qua 1 điện trở Sunt thì đọc điện áp giữa 2 điện trở đó thì lúc đó xài differential này sẽ hợp lý hơn để đo sự sai khác điện áp giữa 2 điểm đó thì cta đo điện áp giữa 2 đầu điện trở shunt thì sẽ ra được cái dòng điện

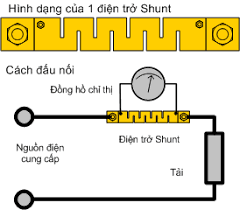
**1. Điện trở Shunt là gì ?**



**Điện trở Shunt là gì**

[**Điện trở shunt**](https://baoanjsc.com.vn/vn/sanpham-17072/Thiet-bi-do-luong/Hioki/Dong-ho-ampe-kim-Hioki-3280-10.aspx) được sử dụng để đo dòng điện, xen kẽ hoặc trực tiếp. Điều này được thực hiện bằng cách đo điện áp rơi qua điện trở. Chúng thường có điện trở nhỏ, được xác định rõ để không ảnh hưởng đến dòng điện đang đo.

**2. Nguyên lý hoạt động điện trở Shunt**



**Nguyên lý hoạt động của điện trở Shunt**

- Lắp [**điện trở shunt**](https://baoanjsc.com.vn/vn/sanpham-16848/Thiet-bi-do-luong/Hioki/Dong-ho-van-nang-hien-thi-kim-Hioki-3030-10.aspx) để tạo ra một sụt áp tỷ lệ với dòng điện đi ngang qua nó, để có thể đo bằng đồng hồ mV. Do đó, các điện trở Shunt được bán ra không phải với trị số điện trở mà thường được ghi với dòng điện định mức, và điện áp tương ứng với dòng định mức đó. Ví dụ Shunt 500A/100mV có nghĩa là dòng định mức 500A, và điện áp rơi trên Shunt khi có dòng 500A đi qua là 100mV

- Cách đấu nối:

**Điện trở Shunt** đấu nối tiếp với tải, dùng các đấu nối lớn (Bu lông lớn), đồng hồ mV nối vào 2 bu lông nhỏ.

- Cách tính toán:

+ Tùy dòng điện cần đo, chọn Shunt có dòng định mức lớn hơn khoảng 50% đến 100%. Theo điện áp giáng định mức, mà chọn đồng hồ mV tương ứng, với một đồng hồ có thang đo tương ứng.

+ Ví dụ: Cần đo dòng điện của một máy nạo bình khoảng 30A, nên mua điện trở Shunt có ghi trị số 50A/100mV. Đồng thời mua một đồng hồ mV có điện áp toàn thang đó (Full scale) FS=100 mV và độ chia từ 0 đến 50 A.

**3. Thông số điện trở Shunt**

- Dòng định mức:  Có sẵn 400A, 600A, 800A, 1000A.

- Ngõ ra: 60mV, 75mV.

- Sai số chính xác: 0.5 phần trăm Full Scale.

- Nhiệt độ hoạt động: -25~60 độ C.

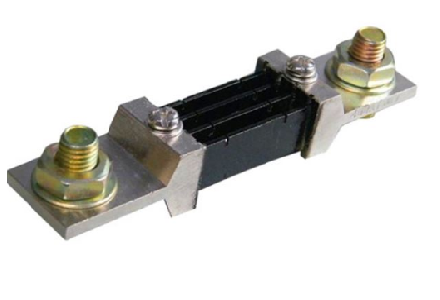
- Khối lượng: 0.78 Kg, tùy vào dòng sơ cấp lớn hay nhỏ sẽ có khối lượng khác nhau.

- Dây sử dụng là 1.5 mét với tiết diện  1.5mm2.

**4. Phân loại điện trở Shunt**

- Đo dòng DC lớn có những dòng định mức sau:

+ [**Điện trở shunt**](https://baoanjsc.com.vn/vn/sanpham-43622/Dung-cu/Hioki/Ampe-kim-do-dong-ro-Hioki-3283.aspx) 400A, 500A, 600A, 700A, 800A, 900A, 1000A



**Điện trở Shunt 400A**

- Đo dòng DC nhỏ có những dòng định mức sau:

+ **Điện trở shunt** 5A, 10A, 20A, 30A, 40A, 50A,...



**Điện trở Shunt 50A**

+**Điện trở shunt** 100A,

+ **Điện trở Shunt** 200A, 250A, 300A

**5. Ứng dụng**

- [**Điện trở shunt**](https://baoanjsc.com.vn/vn/sanpham-30636/Thiet-bi-do-luong/Hioki/Dong-ho-do-dien-tro-Hioki-dong-RM3544.aspx)công dụng bảo vệ mạch chống quá áp

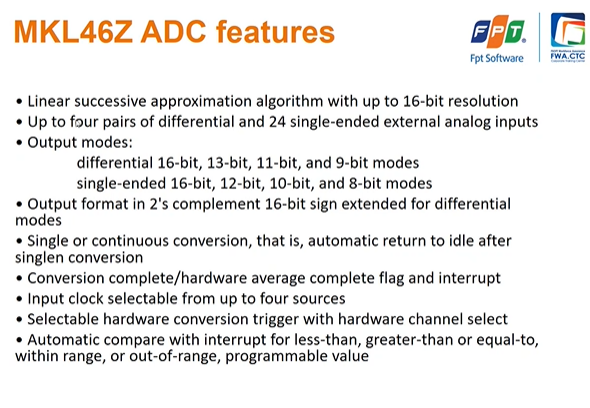
Một phương pháp để bảo vệ mạch điện từ điện áp quá cao là sử dụng mạch điện xà beng. Khi điện áp quá cao, thiết bị sẽ đoản mạch. Điều này dẫn đến dòng chảy song song hiện tại với mạch. Điều này gây ra ngay lập tức một giọt điện áp trong mạch. Dòng điện cao thông qua các shunt nên kích hoạt một bộ ngắt mạch hoặc cầu chì.

- **Điện trở shunt** công dụng bỏ qua một thiết bị bị lỗiKhi một phần tử trong một chuỗi mạch không thành công, nó sẽ phá vỡ mạch hoàn chỉnh. Một shunt có thể được sử dụng để khắc phục vấn đề này. Điện áp cao hơn tồn tại do sự thất bại sẽ khiến cho shunt bị thiếu hụt. Điện sẽ truyền xung quanh yếu tố bị lỗi. Một ví dụ điển hình của việc này là chiếu sáng Giáng sinh.

- **Điện trở shunt** công dụng vượt qua nhiễu điệnShunt với một tụ điện đôi khi được áp dụng trong các mạch mà tiếng ồn tần số cao là một vấn đề. Trước khi tín hiệu không mong muốn đạt đến các phần tử mạch, tụ điện chuyển hướng nhiễu tới mặt đất.

Nếu bạn muốn tìm hiểu về Tự động hóa trong nhà máy, hãy liên hệ với chúng tôi qua **baoanjsc@gmail.com**. Với phương châm làm việc chuyên nghiệp, tận tâm Bảo An Automation luôn cam kết mang tới cho khách hàng sản phẩm với chất lượng tốt nhất với giá thành hợp lý và đảm bảo giao hàng đúng tiến độ.

Còn signle – end này thì là 1 chân đọc so sánh với cái điện áp tham chiếu GND của mình



ở trên này có 4 cặp differential, 24 chân signle

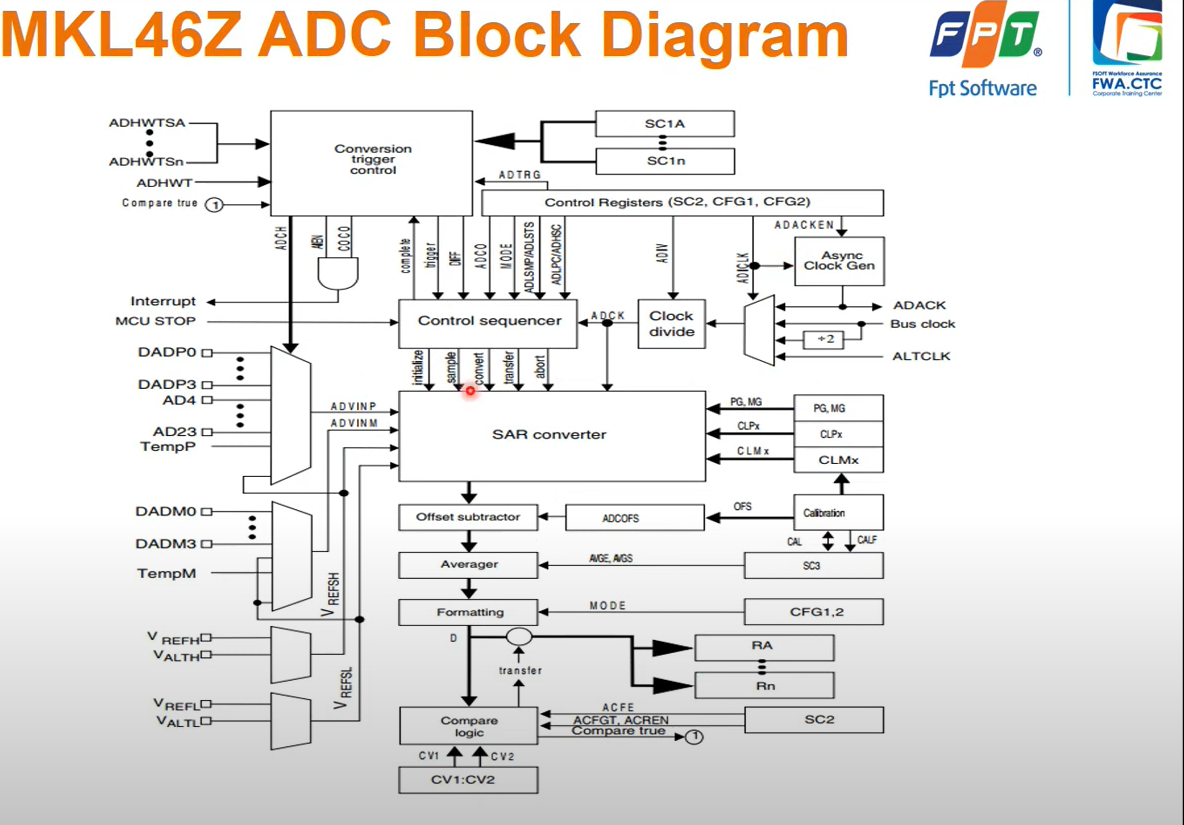
ở mode differential thì có thể cấu hình nó là 16 bit, ,13 bit, 11 bit hay 9 bit. Còn ở single có thể cấu hình 16 12 10 và 8 bit.

Làm đồ án với STM32

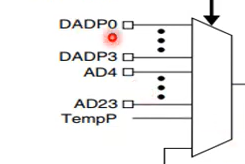
Đọc biến trở volume, đọc điện áp từ cầu phân áp của biến trở 10K

ở ứng dụng đó cấu hình độ phân giải 16 bit. Stmf1btt16 ở độ phân giải 16 bit, bước nhảy là 3,3/2^16.

Adc trong kl46 có thể đọc trong datasheet của nó

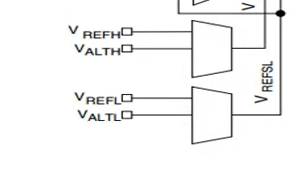


Các ô hình thang là các nấc chọn đầu vào



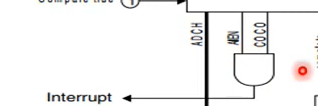
Ví dụ có 4 cái differential ở đây thì có thể đo từ DADP0 đến DADP3 là differential

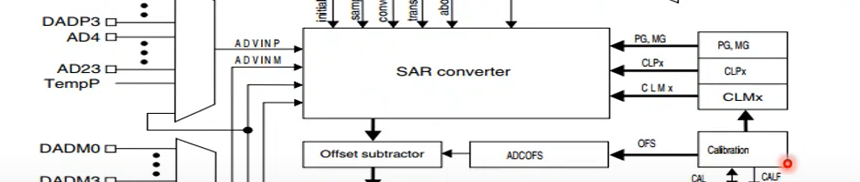
Chọn từ 0 tới 23 là single rồi clock thì chọn clock nào.



ở đây là chọn các thằng nào

đọc trong datasheet để chọn

ở đây interrrupt thì cta  cài đặt cho nó, nếu chuyển đổi xong thì có ngắt hoặc là ko thì tuỳ cta cấu hình

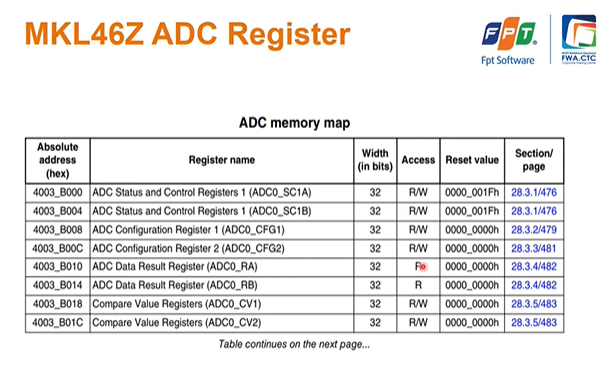


ở đây sẽ thấy có calib. Tuy nhiên là có những con hỗ trợ bằng cách tự bù bằng tụ rồi cấu hình low power nghĩa là tiêu thụ năng lượng thấp thì ở trong con MCU có nhiều module, có phần điều phối, có phần timer ADC các thứ

có những khối hoạt động ko cần tới core như timer (timer chỉ cần có clock là nó đếm thôi) và những khối khác

trong core rất là phức tạp (vì tích hợp nhiều cách), để core hoạt động thì tiêu tốn rất nhiều năng lượng. có những ứng dụng ko cần tới core thì đưa nó vào mode lower để nó chỉ hoạt động mỗi cái module mà cta cần thôi (nghĩa là analog thôi), analog biến đổi xong core ko cần đọc kết quả làm gì hết thì nó cứ thế biến đổi và 1 module/ khối khác đọc kết quả rồi nó ghi thẳng vào bộ nhớ hay địa chỉ bên ngoài.

Nghĩa là 1 khối nào đó hoạt động với analog mà ko cần tới core thì ngta sẽ đưa nó vào mode lower để tiết kiệm năng lượng hoặc là ngta sẽ đưa vào 1 mode low speed để nó hoạt động ở tốc độ chậm hơn thì tiết kiệm năng lượng hơn



Các memory map thì tự tìm hiểu

Tổng kết: tập trung vào 3 kiến trúc: flash, biến đổi xấp xỉ và ramp compare thì ramp compare khắc phục nhược điểm của biến đổi xấp xỉ vì nó đếm từ giá trị trước đó của nó chứ ko đếm từ 0 lên

Còn flash thì ưu điểm là nhanh, nhược điểm là tiêu tốn tài nguyên lớn nên ngta ít ứng dụng flash, chỉ đời cũ thôi

Hiện này ngta sử dụng 2 cái còn lại

Có rất nhiều bộ chuyển đổi ADC, mình chỉ tìm hiểu 3 cái

