

Thiết kế hệ thống nhúng-EE4251

GV: ĐÀO ĐỨC THỊNH

KHOA TỰ ĐỘNG HOÁ-TRƯỜNG ĐIỆN ĐIỆN TỬ-ĐHBK HN

A solid orange horizontal bar spanning the width of the slide, located at the bottom.

Phần cứng hệ nhúng

- Ngoại vi cơ bản
- Giao tiếp với tín hiệu Analog
- Truyền tin
- Bộ nhớ
- CPU
- Quản lý năng lượng
- Phần cứng bảo mật.
- Các vấn đề khác

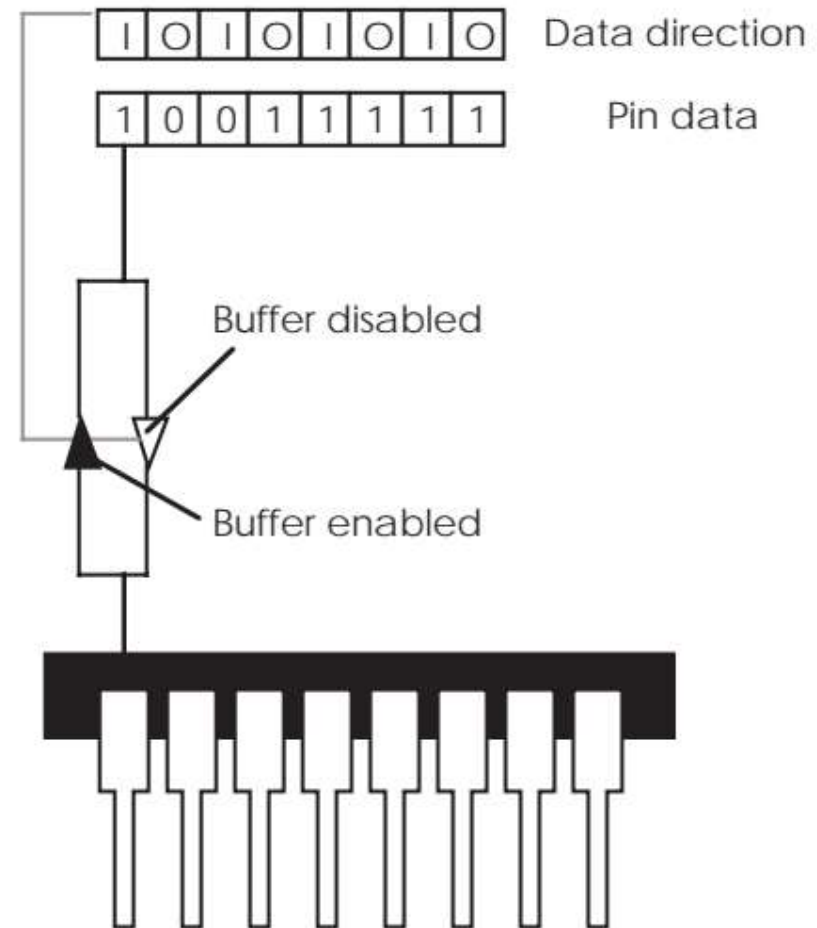
Các ngoại vi cơ bản

Cổng song song:

- Truy xuất dữ liệu một bit cho mỗi chân tín hiệu.
- Đọc/ghi đồng thời nhiều chân tín hiệu (8,16..) cùng một lúc.
- Cũng có thể truy nhập đến từng bit riêng.
- Giao tiếp với máy in cổng Centronics, LED, màn hình số. Đầu vào công tắc, bàn phím bằng ĐK...
- Cơ bản là thanh ghi 8 bits.
- Đầu vào được lấy mẫu tín hiệu tại thời điểm đọc
- Đầu ra có thanh ghi chốt.

Các ngoại vi cơ bản

Cấu trúc cổng vào/ra song song:

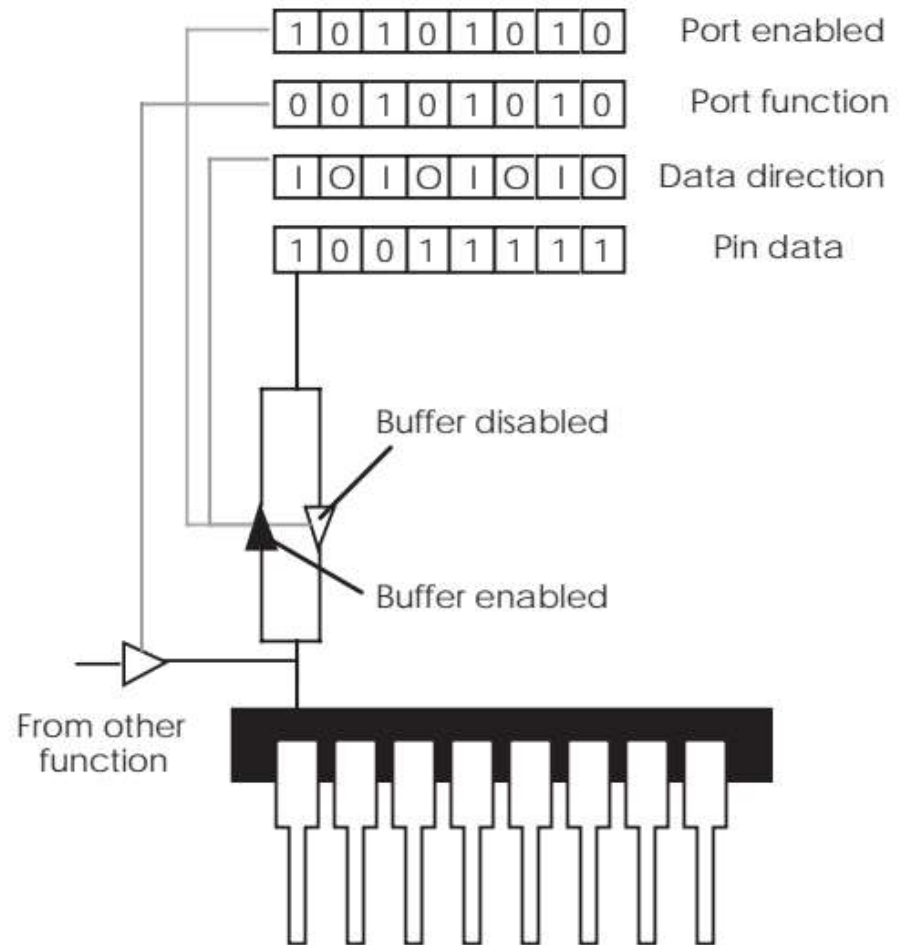


Các ngoại vi cơ bản

- Nó là nền tảng của nhiều chip giao diện song song đời đầu như Intel 8255 và Motorola 6821.
- Các cổng đầu ra có thể thực hiện được điều này thường được gọi là cổng ba trạng thái.
- Một số cổng có thể bổ xung các chân tín hiệu giao tiếp

Các ngoại vi cơ bản

Cổng đa chức năng:



Các ngoại vi cơ bản

- Các chân I/O có thể chia sẻ với thiết bị ngoại vi khác (timer...)
- Việc thiết lập chức năng được thực hiện bằng các thanh ghi.
- Số lượng thực tế chân I/O giảm đi nhiều khi thực hiện đa chức năng.
- Một số thiết bị cần có R Pull-up để tránh hoạt động sai của cổng.

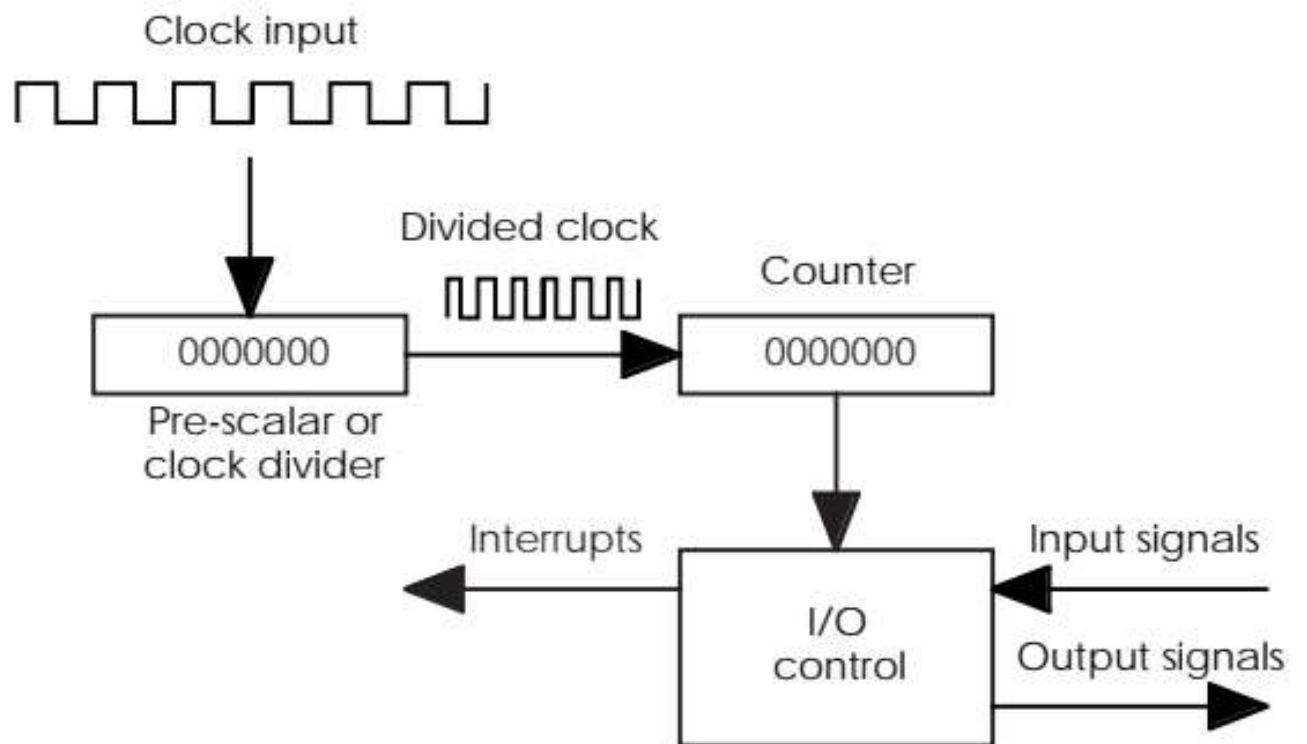
Các ngoại vi cơ bản

Timer/counter:

- T/C sử dụng để cung cấp thời gian hay đếm sự kiện với yêu cầu xử lý phần mềm tối thiểu
- Hầu hết hệ nhúng đều sử dụng. Tạo thời gian cho phần mềm, cung cấp nhịp cho HĐH, baud rate, âm thanh..
- Bộ đếm có thể 8, 16, 24 bit
- Tạo ra các tần số bởi tần số đầu vào và giá trị bộ đếm pre-scalar. Giá trị tối thiểu quyết định bởi tần số của tín hiệu vào bộ đếm.
- Kích thước bộ đếm xác định giá trị tối đa.
- @ thông số quyết định sự phù hợp của T/C với 1 ứng dụng

Các ngoại vi cơ bản

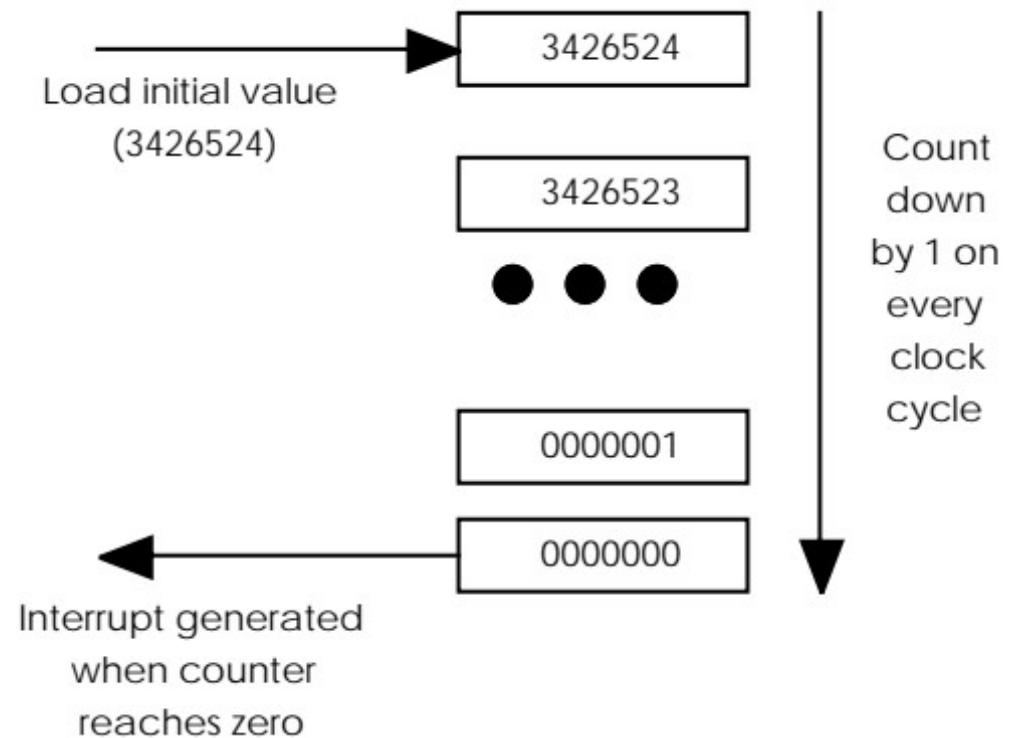
Cấu trúc T/C:



Các ngoại vi cơ bản

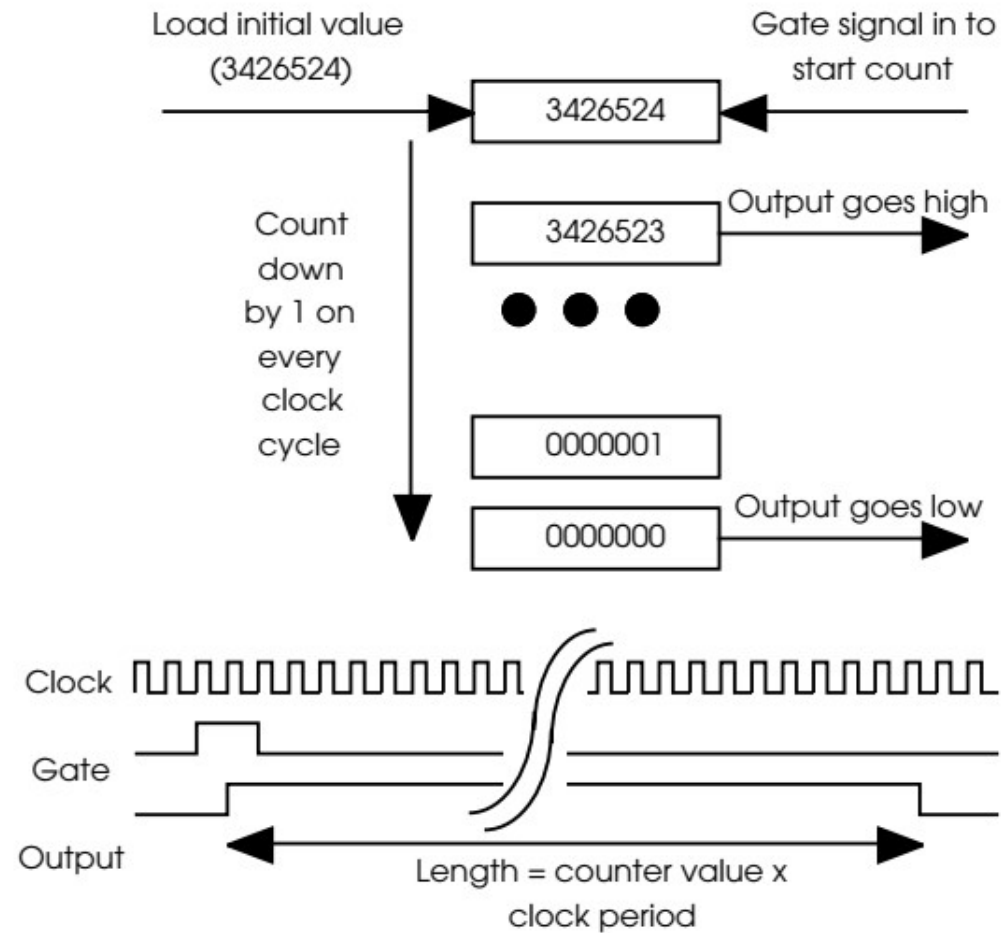
Intel 8252

- Có 3 bộ đếm.
- Cung cấp tick cho hệ thống 15us để làm tươi RAM
- Chế độ ngắt: nạp trước một giá trị đếm về 0 tạo ra ngắt



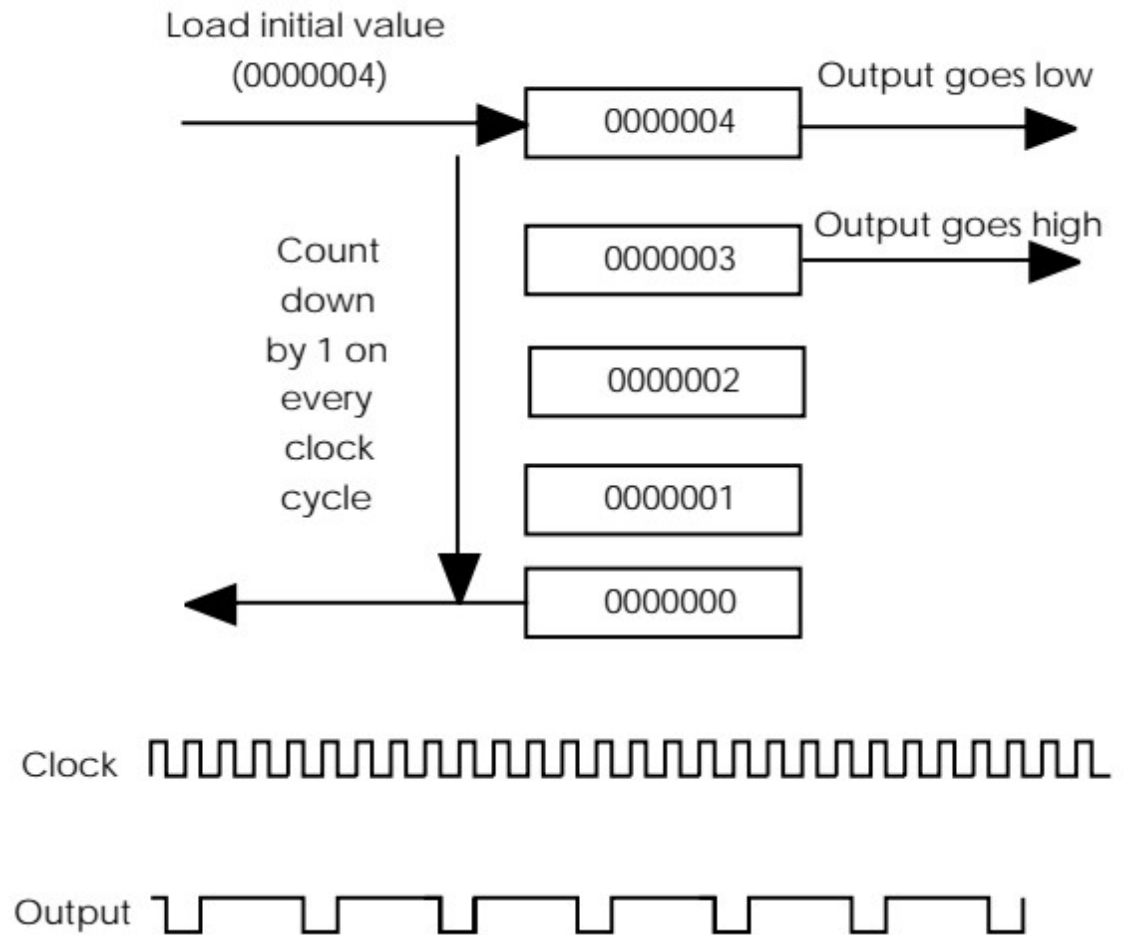
Các ngoại vi cơ bản

- One-shot: tạo ra một xung ngắn lập trình tự sự kiện đầu vào.



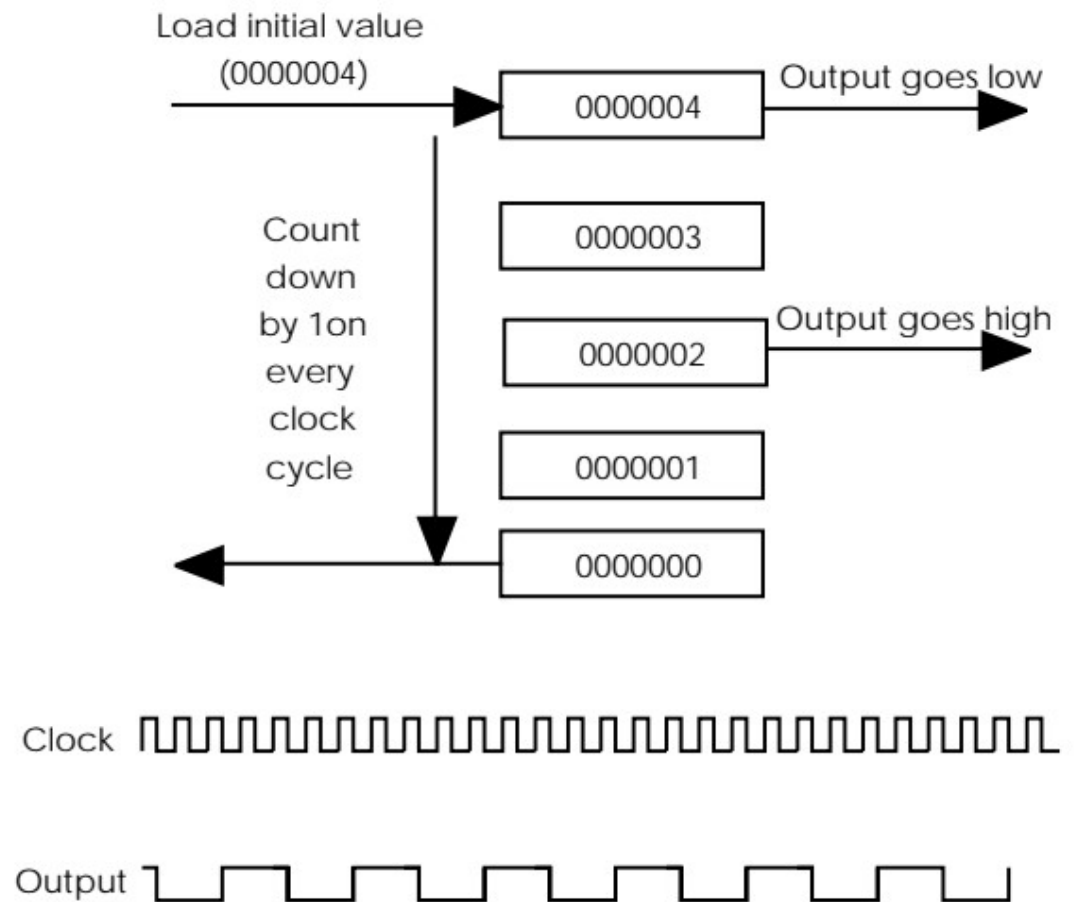
Các ngoại vi cơ bản

- Chế độ phát xung chia N:



Các ngoại vi cơ bản

Chế độ phát xung vuông:



Các ngoại vi cơ bản

Time processor: hiện các hệ thống nhúng sử dụng nhiều clock với mục đích khác nhau như cấp cho CPU, bộ nhớ, I/O các ngoại vi và các hệ thống khác. Một bộ TP sẽ sử dụng trong trường hợp này.

Các ngoại vi cơ bản

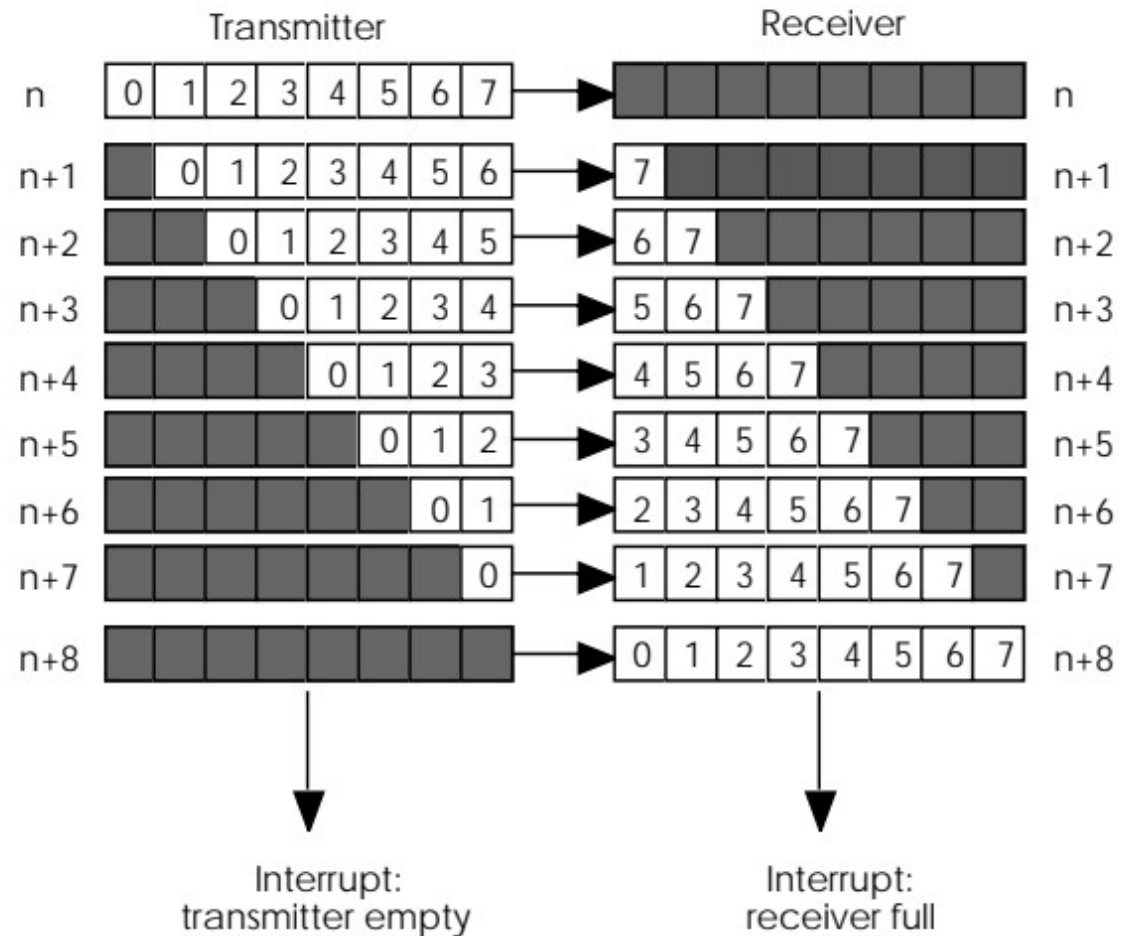
Real-time Clock:

- Đây là bộ đếm thời gian đặc biệt cung cấp thời gian thực và ngày tháng
- Sử dụng IC đặc biệt
- Thạch anh 32.768 kHz
- Có nguồn nuôi bằng pin hay Super-CAP để hoạt động khi không có nguồn

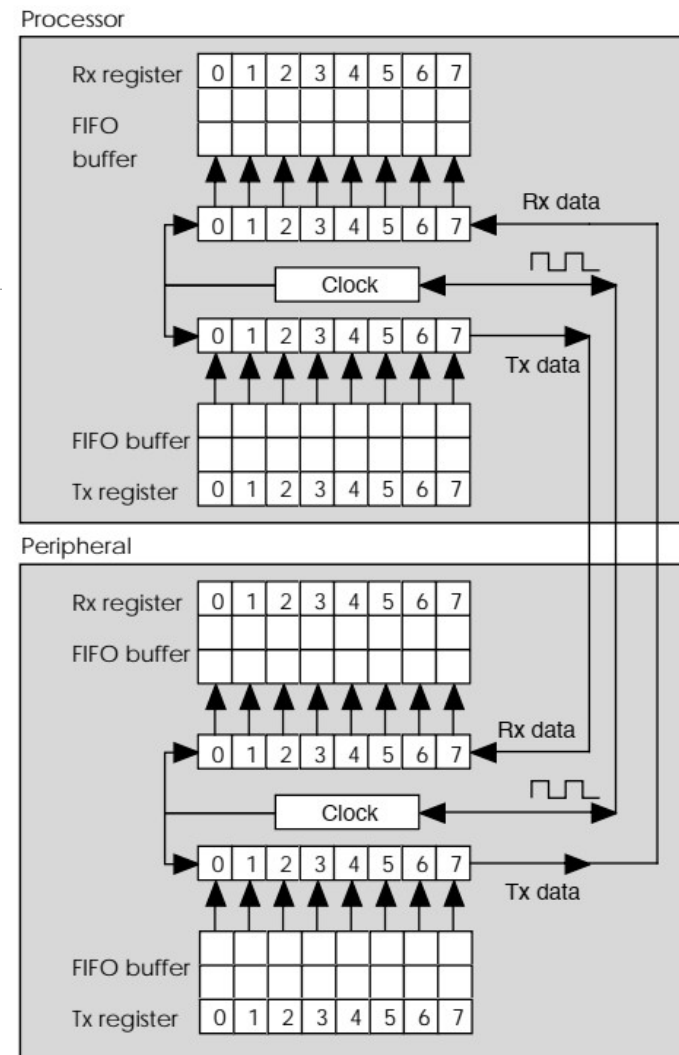
Các ngoại vi cơ bản

Cổng nối tiếp:

- Nguyên lý hoạt động cơ bản của cổng nối tiếp



Các ngoại vi cơ bản

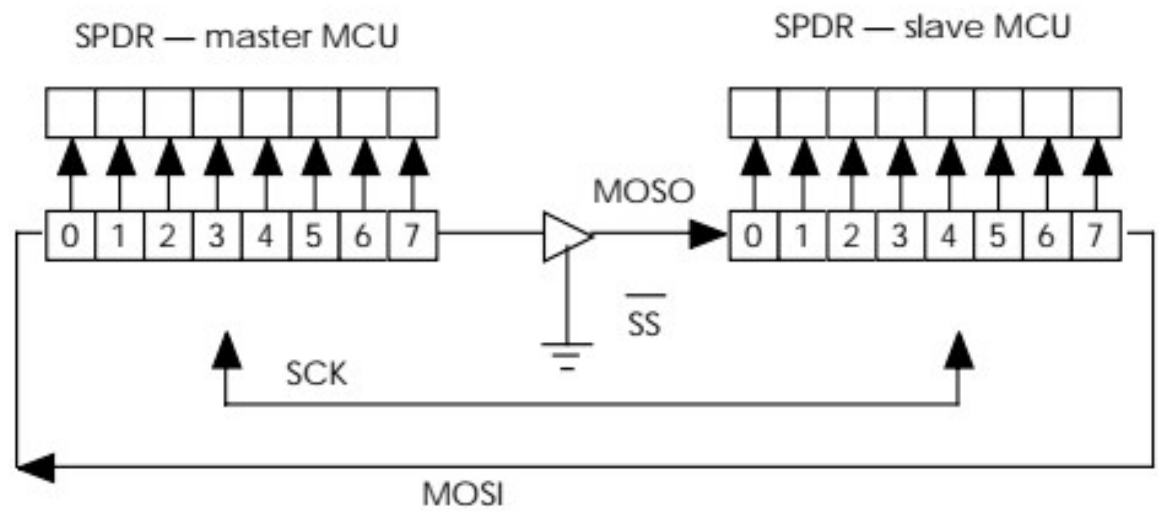


Các ngoại vi cơ bản

- Sử dụng một cặp thanh ghi dịch
- Nối với nhau
- Sử dụng một Clock chung
- Dữ liệu truyền từ thanh ghi phát sang thanh ghi nhận
- Thời gian truyền phụ thuộc tần số clock
- Các thanh ghi thường 8 bit
- Khi kết thúc có thể tạo ra các ngắt.
- Có thể có thêm các bộ nhớ đệm FIFO

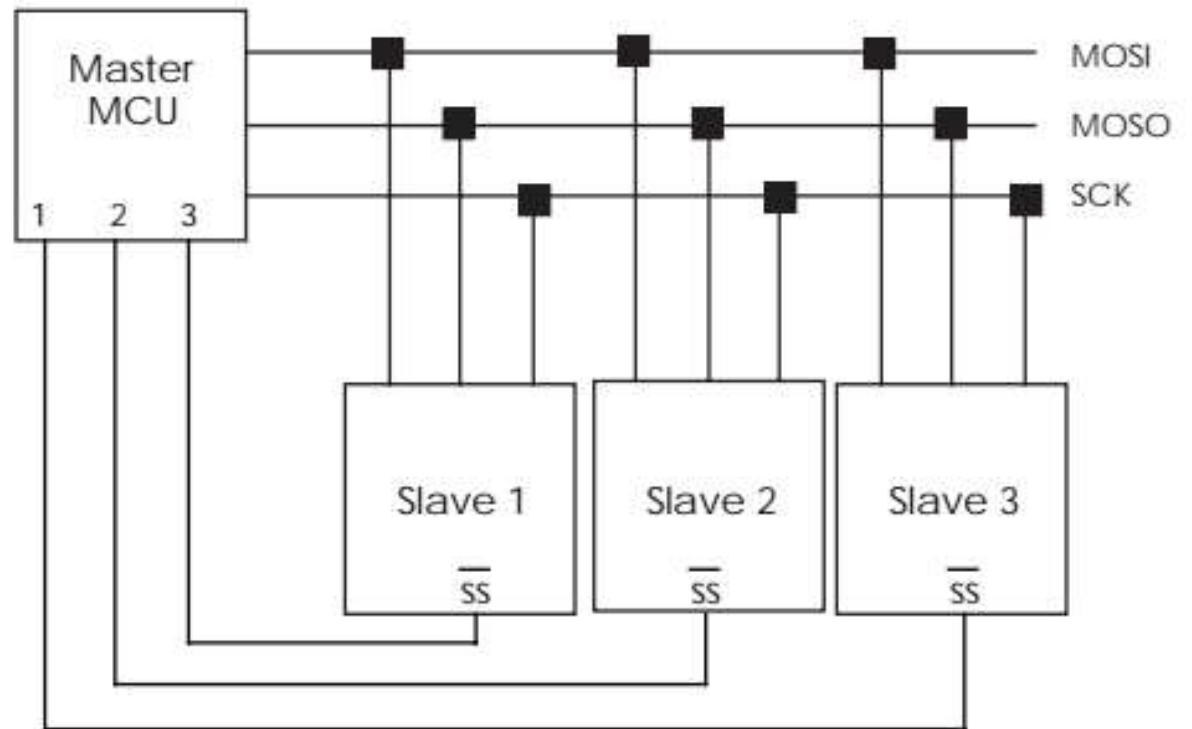
Các ngoại vi cơ bản

Cổng SPI



Các ngoại vi cơ bản

SPI với nhiều Slave

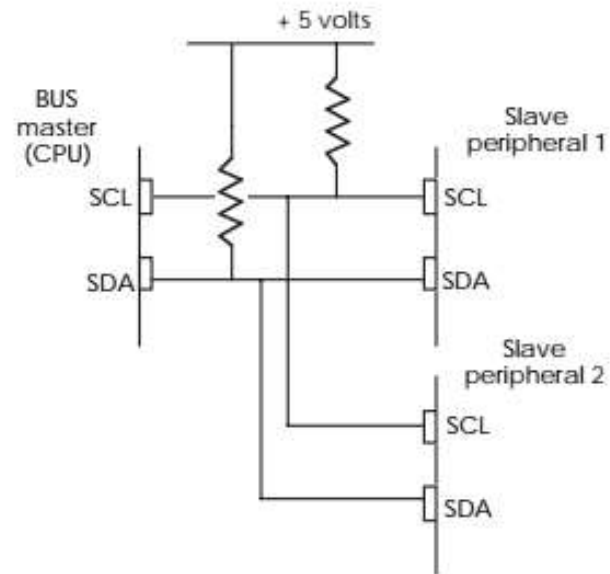


Các ngoại vi cơ bản

- Inter-IC hay I2C
- Bus chỉ gồm 2 tín hiệu SDA và SCL
- Với số lượng nhỏ thiết bị có thể gắn trực tiếp.
- Nếu số lượng lớn có thể dùng bộ đệm
- Nó có thể kết nối một vài thiết bị với khoảng cách lên đến m.

Các ngoại vi cơ bản

Kết nối mạch điện

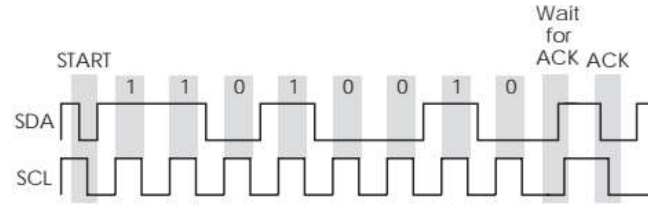


Các ngoại vi cơ bản

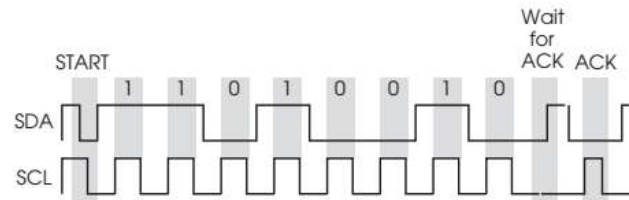
- Các bit của IIC

Message	1st event	2nd event
START	SDA H\L	SCL H\L
STOP	SCL L\H	SDA L\H
ACK	SDA H\L	SCL H\L

- Ghi một byte với ACK

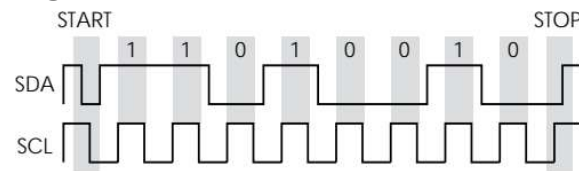


- Đọc một byte với ACK

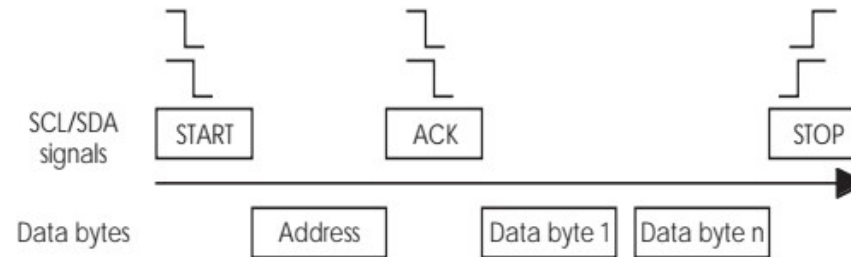


Các ngoại vi cơ bản

- Ghi một byte với STOP



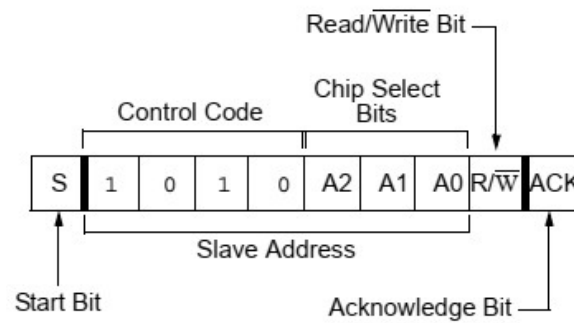
- Kết nối nhiều thiết bị nên slave được chọn bởi byte đầu tiên. Nếu thiết bị đúng địa chỉ nó gửi ACK không thì SDA không đổi.



Các ngoại vi cơ bản

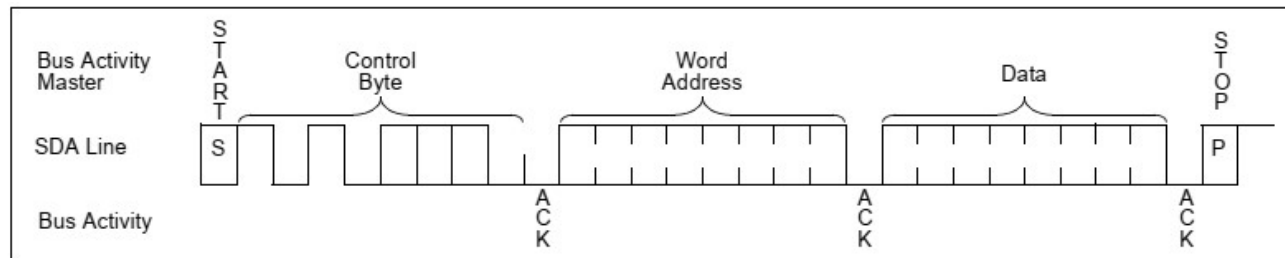
Ví dụ cho 24C01

- Byte điều khiển

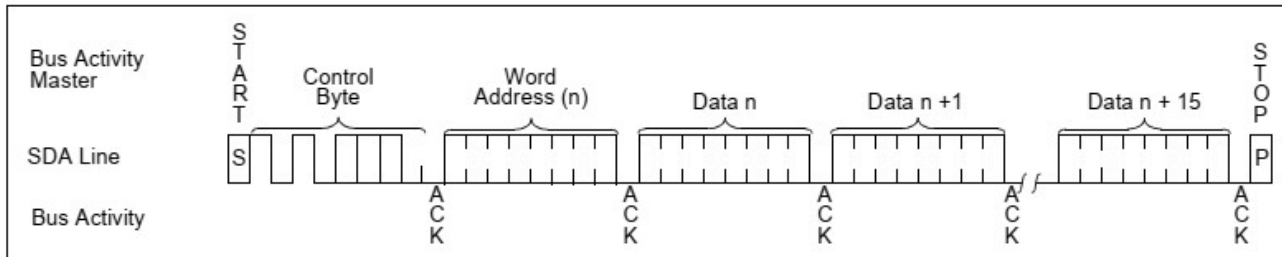


Các ngoại vi cơ bản

- Ghi một byte:

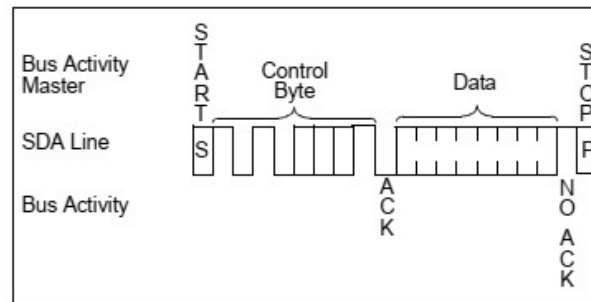


- Ghi một trang

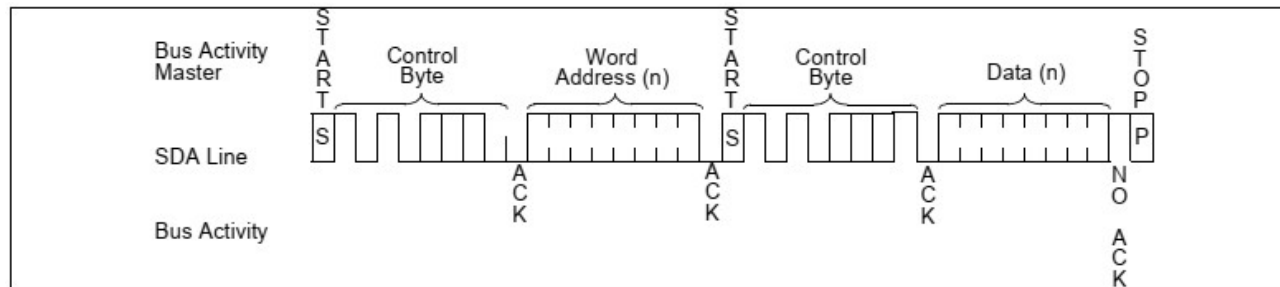


Các ngoại vi cơ bản

- Đọc tại vị trí hiện tại

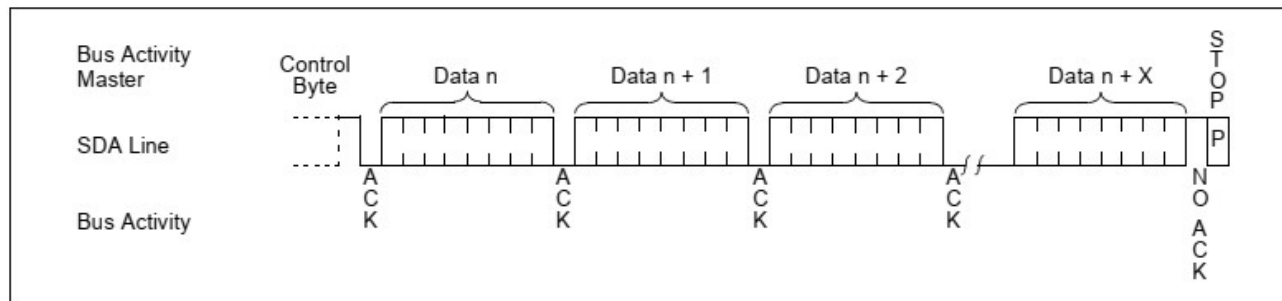


- Đọc ngẫu nhiên



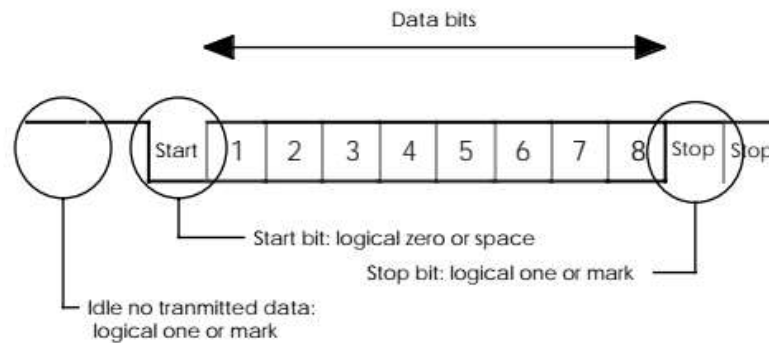
Các ngoại vi cơ bản

- Đọc tuần tự



Các ngoại vi cơ bản

- Truyền thông không đồng bộ, định dạng một ký tự



Các ngoại vi cơ bản

Các chuẩn vật lý

- RS-232
- RS-422
- RS-485

Các ngoại vi cơ bản

- Điều khiển lưu lượng truyền:

DB-25	Signal	DB-9
1	Chassis ground	Not used
2	Transmit data — <i>TXD</i>	3
3	Receive data — <i>RXD</i>	2
4	Request to send — <i>RTS</i>	7
5	Clear to send — <i>CTS</i>	8
6	Data set ready — <i>DSR</i>	6
7	Signal ground — <i>GND</i>	5
8	Data carrier detect — <i>DCD</i>	1
20	Data terminal ready — <i>DTR</i>	4
22	Ring indicator — <i>RI</i> or <i>RING</i>	9

Các ngoại vi cơ bản

- Điều khiển lưu lượng XON/XOFF
- XOFF sẽ được bên nhận gửi báo cho bên phát không gửi cho đến khi nhận được XON.

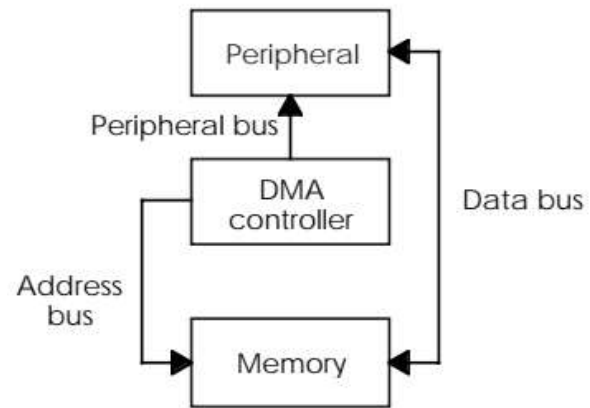
Các ngoại vi cơ bản

DMA controllers

- DMA giải pháp phần cứng cung cấp phương pháp truyền dữ liệu hiệu quả từ ngoại vi đến bộ nhớ
- Khi không có DMA uP dùng polling để thực hiện việc trao đổi, Việc này tốn nhiều xử lý và có thể bị mất dữ liệu
- Ngắt là một giải pháp tốt hơn nhưng nó dừng công việc hiện tại lại. Một cản trở thời gian quay trở lại công việc đang làm cũng như việc lưu trữ các thanh ghi và bộ nhớ đệm
- DMA thực hiện trao đổi giả ngoại vi với bộ nhớ hay 2 bộ nhớ với nhau. DMA là một phần cứng không yêu cầu xử lý từ uP

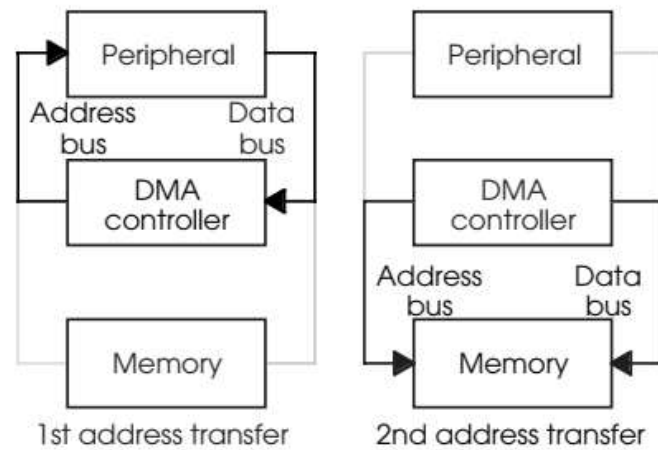
Các ngoại vi cơ bản

- Chế độ địa chỉ đơn



Các ngoại vi cơ bản

- Chế độ truyền địa chỉ kép



Giao tiếp với Analog

Sensor

- Cảm biến có thể được thiết kế cho hầu hết mọi đại lượng vật lý. Có các cảm biến về trọng lượng, vận tốc, gia tốc, dòng điện, điện áp, nhiệt độ, v.v.
- Các hiệu ứng có thể cảm ứng điện từ, quang điện, hoá điện...
- Cảm biến đã đạt được những tiến bộ lớn

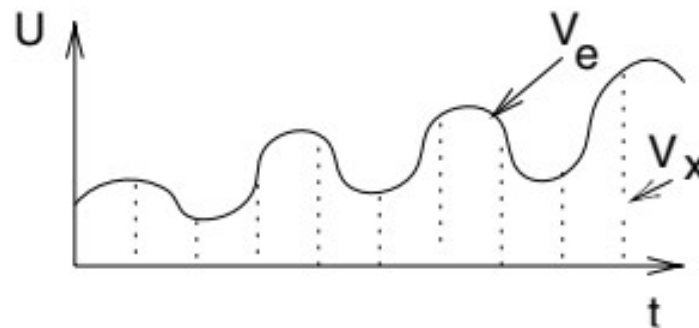
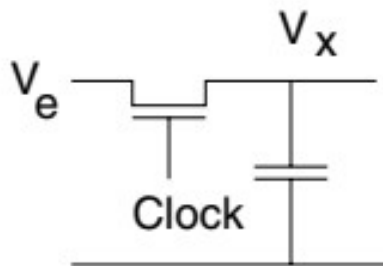
Giao tiếp với Analog

- Cảm biến gia tốc
- Cảm biến hình ảnh CCD
- Cảm biến trắc sinh học
- Mắt nhân tạo
- RFID
- Cảm biến trong ứng dụng Ô-tô.
- Các cảm biến khác: như cảm biến nhiệt, Hall sensor....

Giao tiếp với Analog

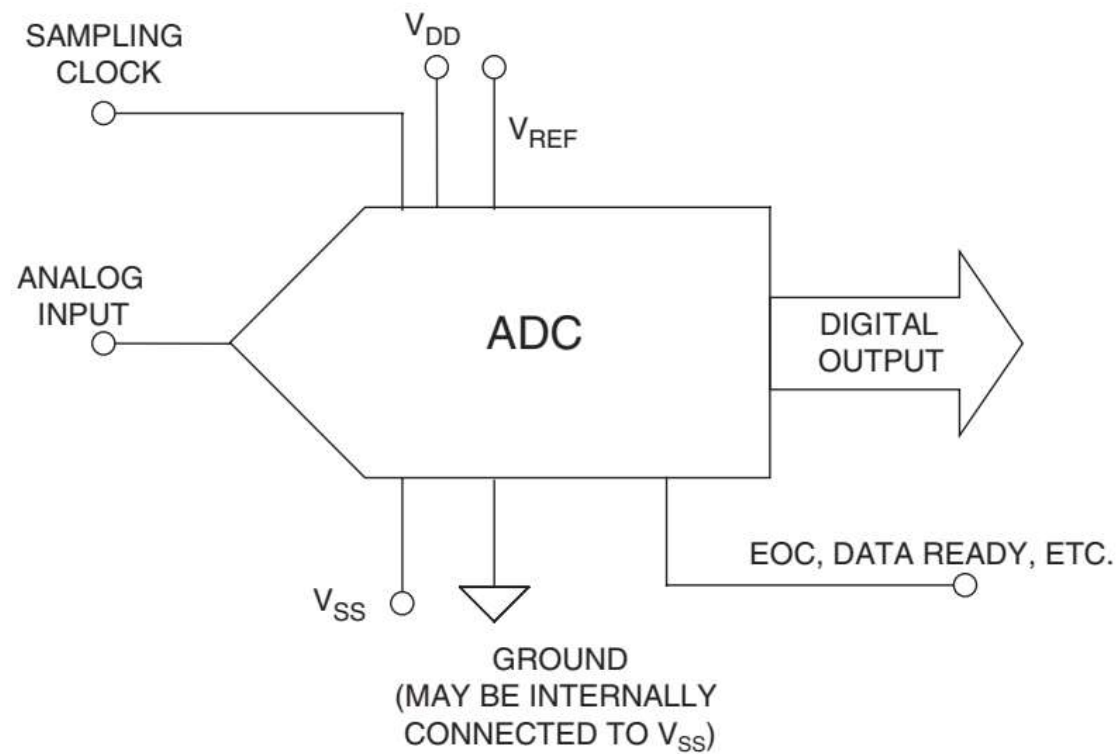
Mạch ghi và giữ mẫu:

- Máy tính được hoạt động trong miền thời gian rời rạc.
- Tín hiệu cần được lấy mẫu trong miền thời gian



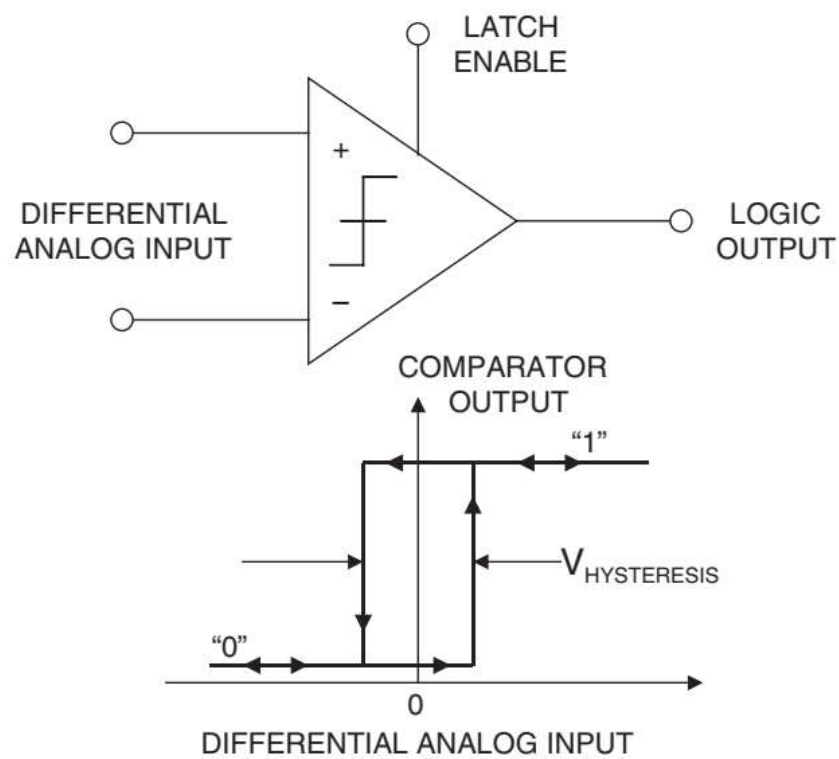
ADC

Cấu trúc cơ bản của ADC



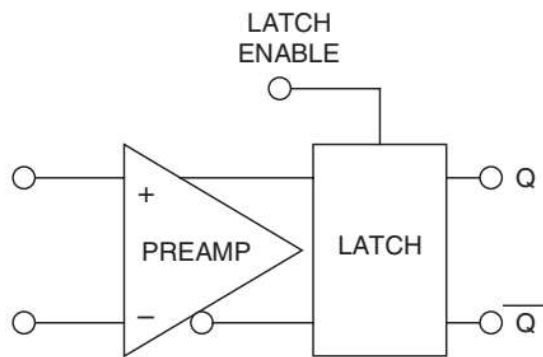
ADC

Mạch so sánh: 1 bit ADC

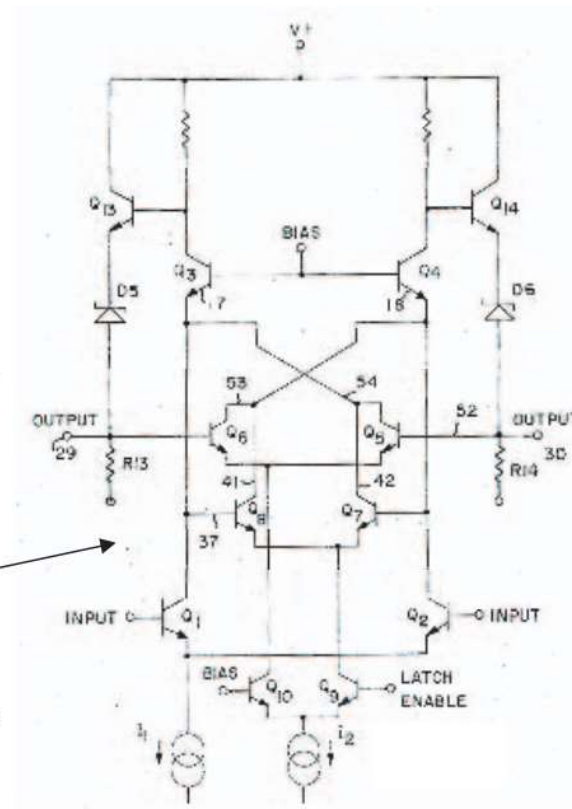


ADC

AM685



From James N. Giles, "High Speed Transistor Difference Amplifier," U.S. Patent 3,843,934, filed January 31 1973, issued October 22, 1974

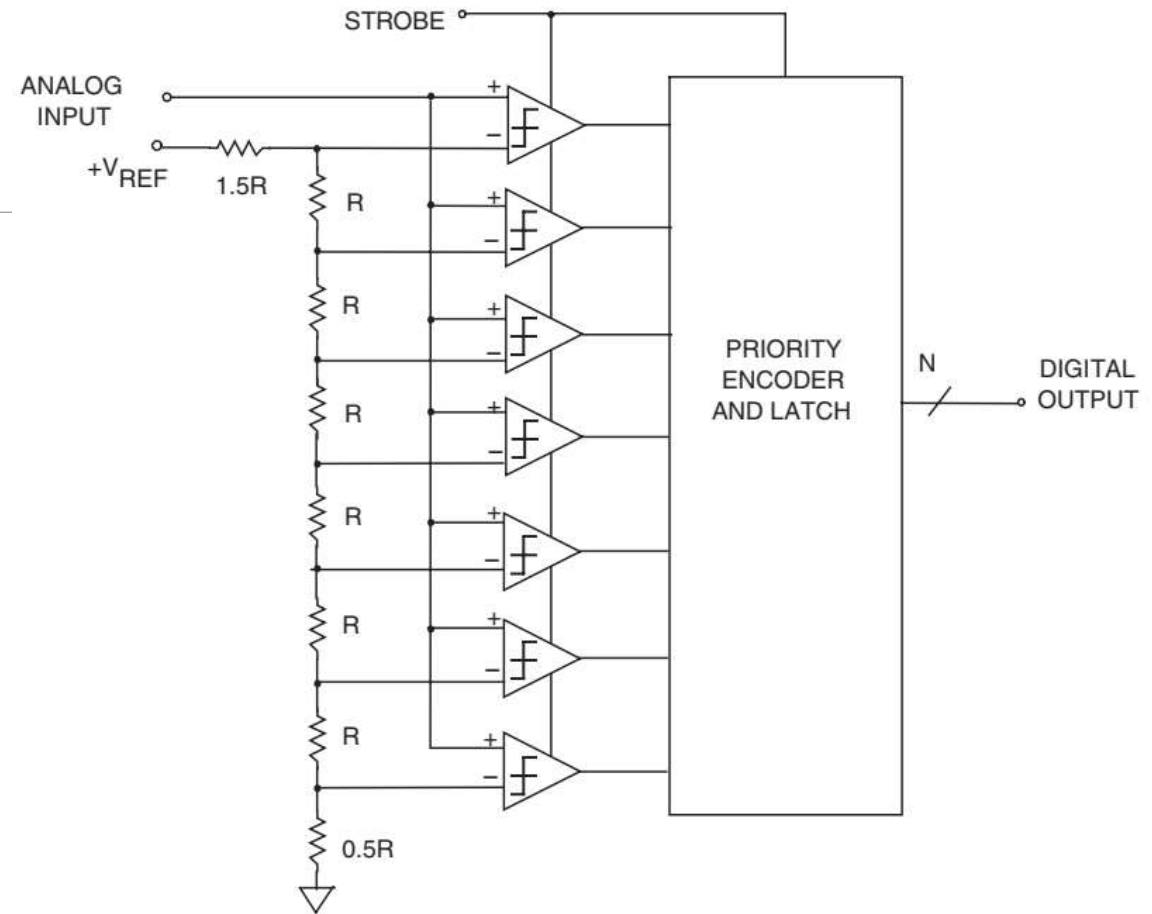


ADC

ADC tốc độ cao: Flash ADC

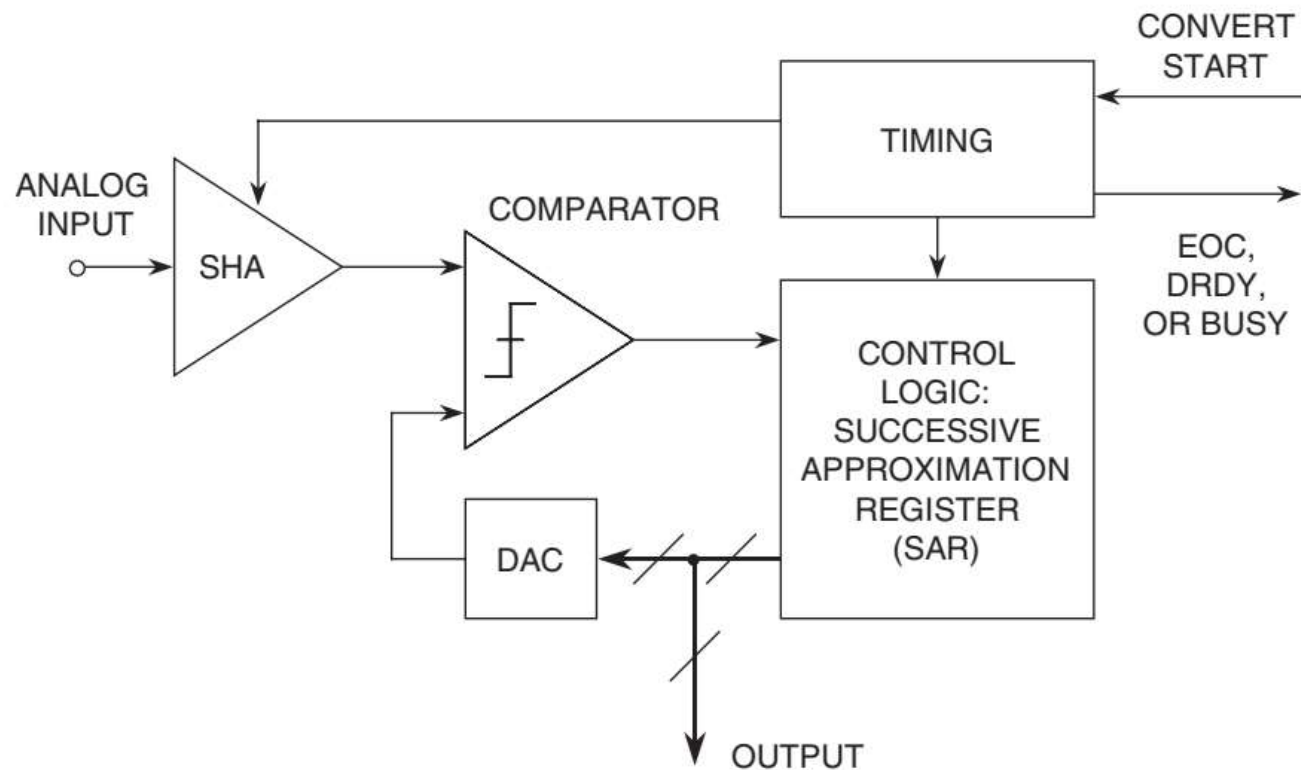
Flash ADC có tốc độ lấy mẫu $>50\text{MSPS}$

Thường 6-8-10 bit.

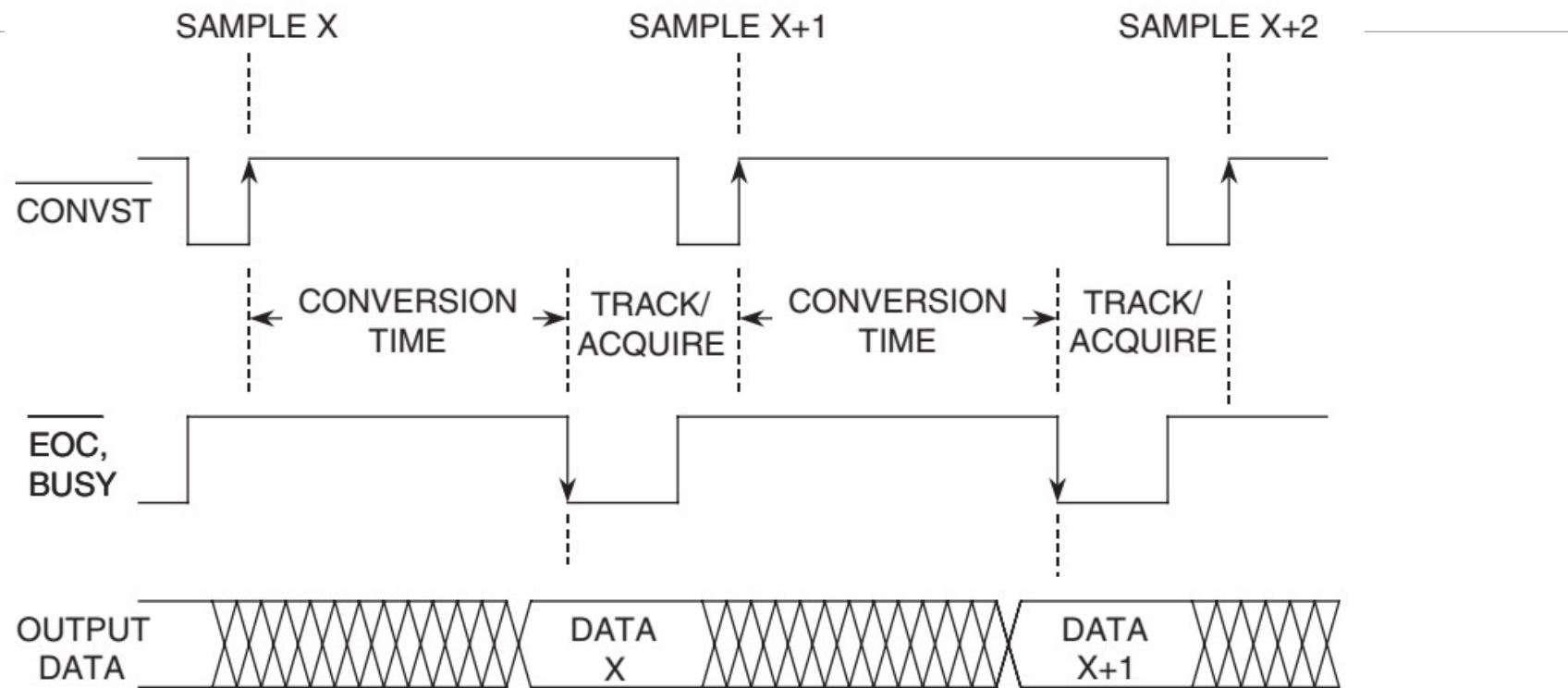


ADC

ADC xấp xỉ gần đúng liên tiếp:



ADC

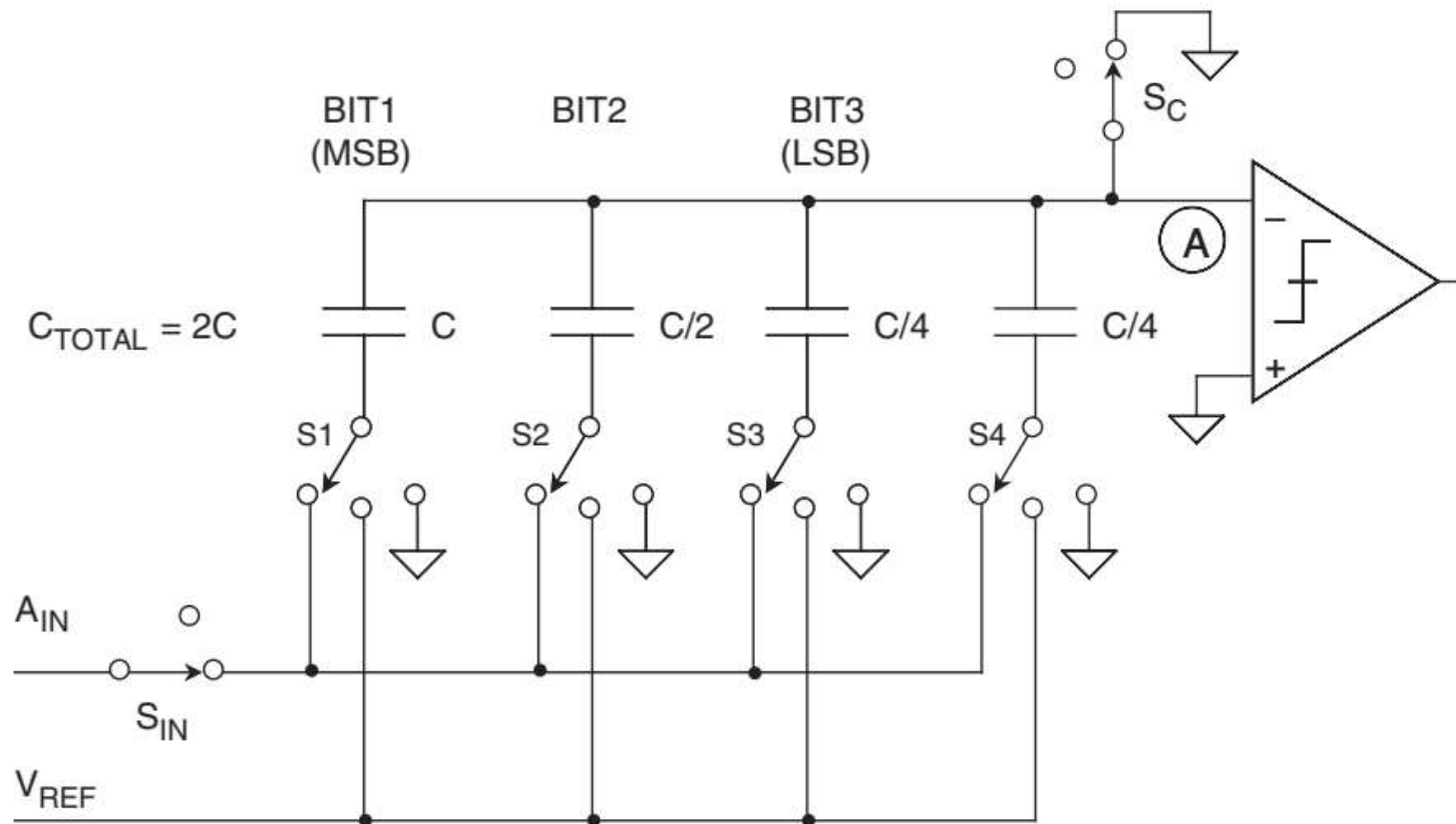


ADC

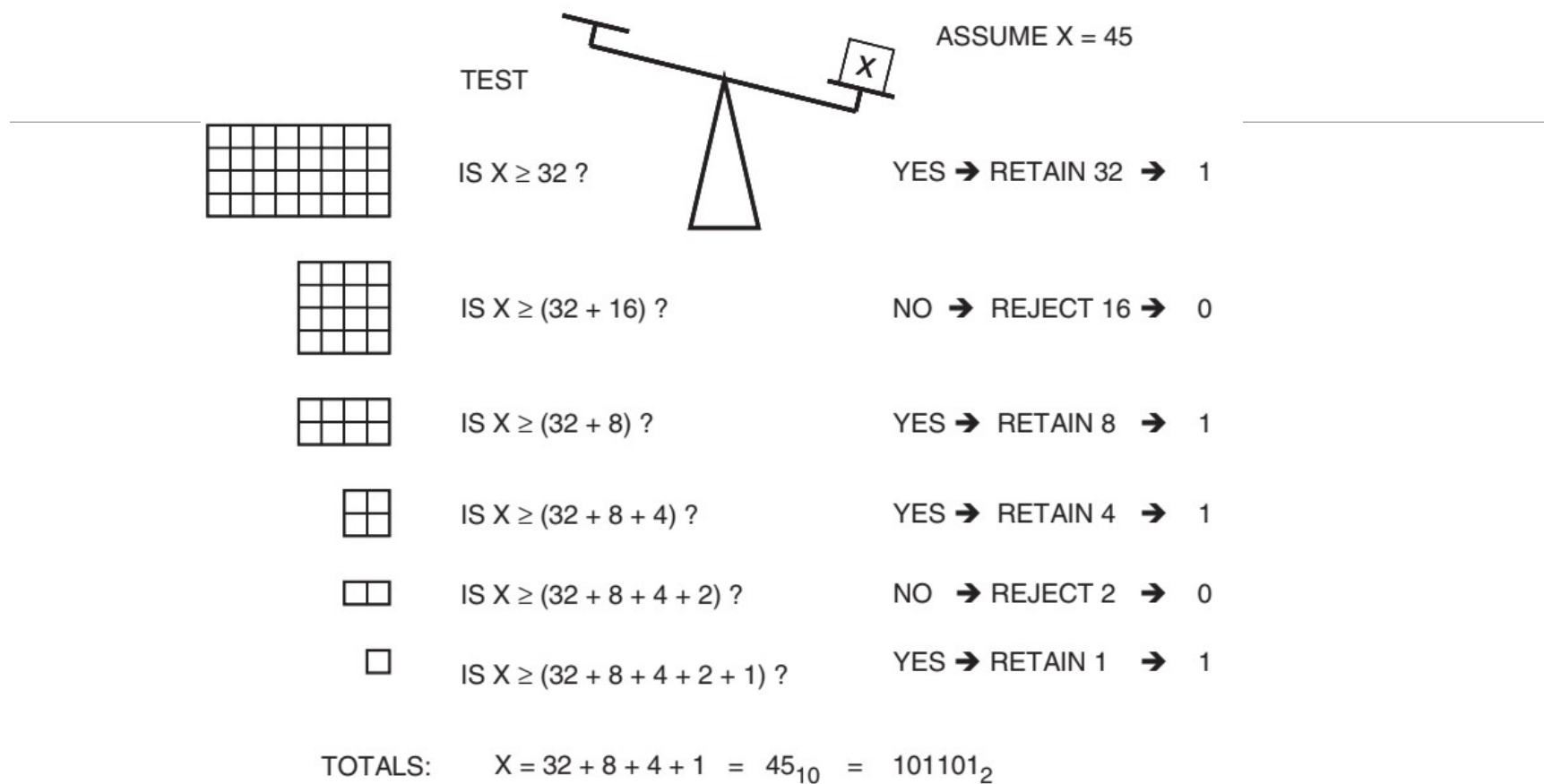
Độ phân giải 8-16 bit.

Thời gian biến đổi: 8 bit vài trăm ns, 16 bit vài us.

ADC



ADC

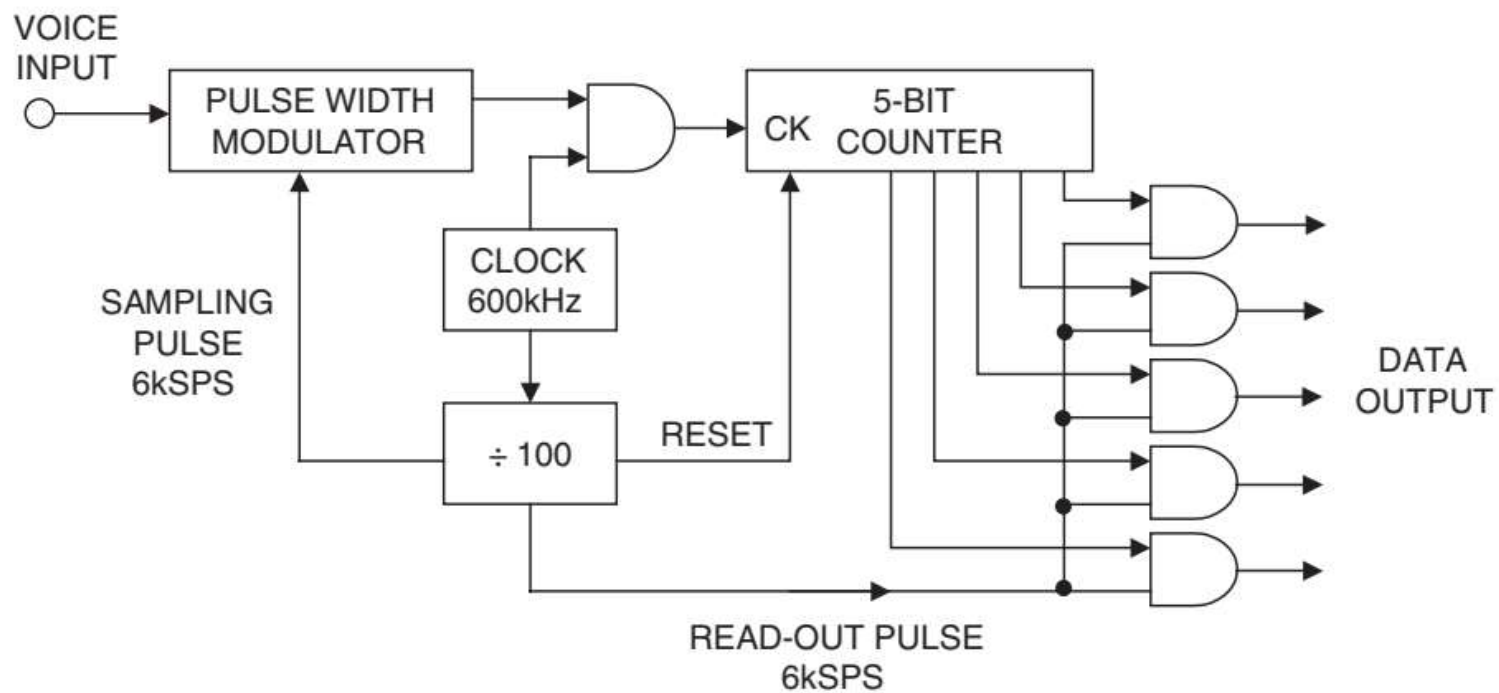


ADC

	RESOLUTION	SAMPLING RATE	POWER	CHANNELS
AD7482	12 BITS	3.0MSPS	80mW	1
AD7484	14 BITS	3.0MSPS	80mW	1
AD7490	12 BITS	1.0MSPS	6mW	16
AD7928	12 BITS	1.0MSPS	5.4mW	8
AD974	16 BITS	0.2MSPS	120mW	4
AD7677*	16 BITS	1.0MSPS	130mW	1
AD7621*	16 BITS	3.0MSPS	100mW	1
AD7674*	18 BITS	0.8MSPS	120mW	1

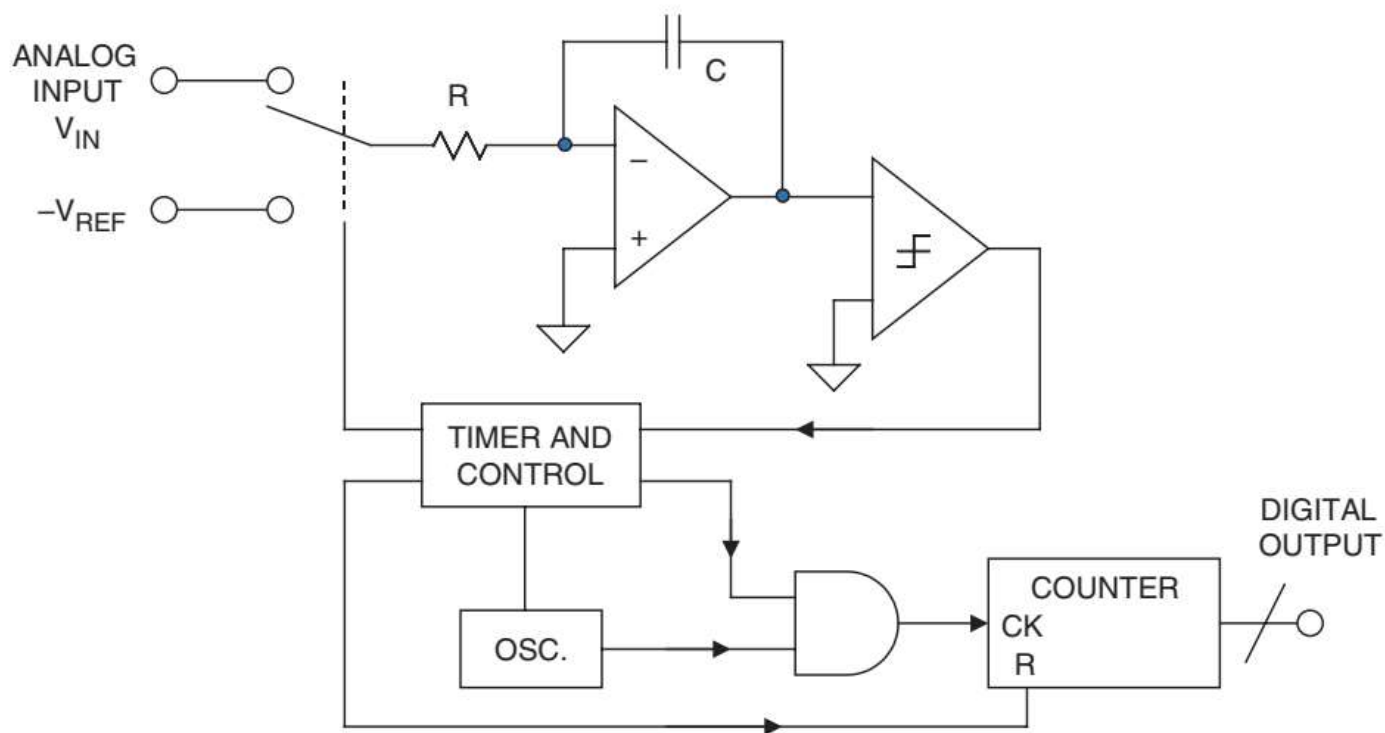
ADC

ADC kiểu đếm:

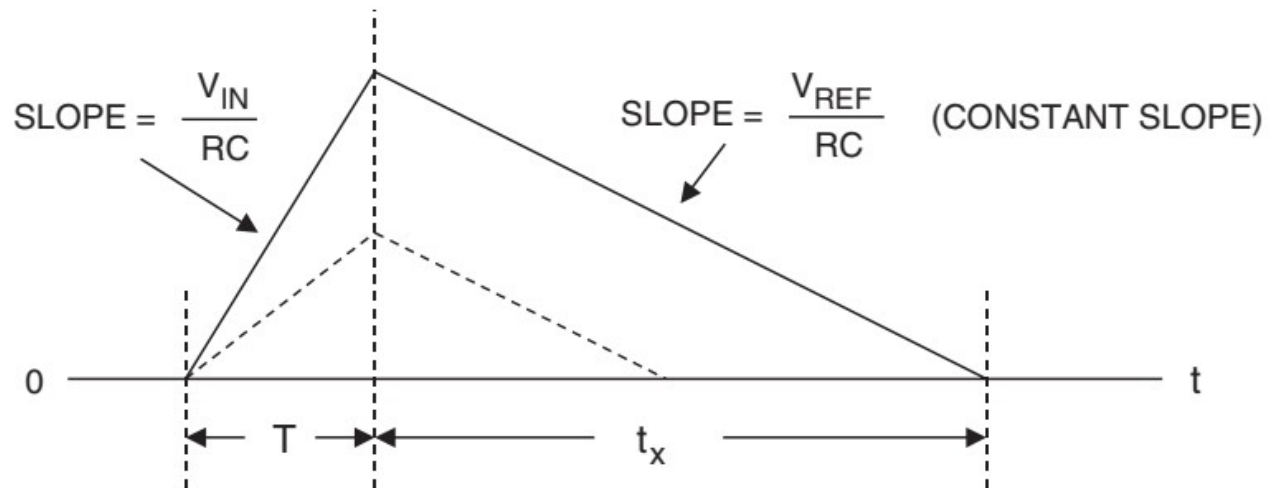


ADC

ADC tích phân 2 sườn xung:



ADC



$$\frac{V_{IN}}{RC} T = \frac{V_{REF}}{RC} t_x$$

$$t_x = \frac{V_{IN}}{V_{REF}} T$$

ADC

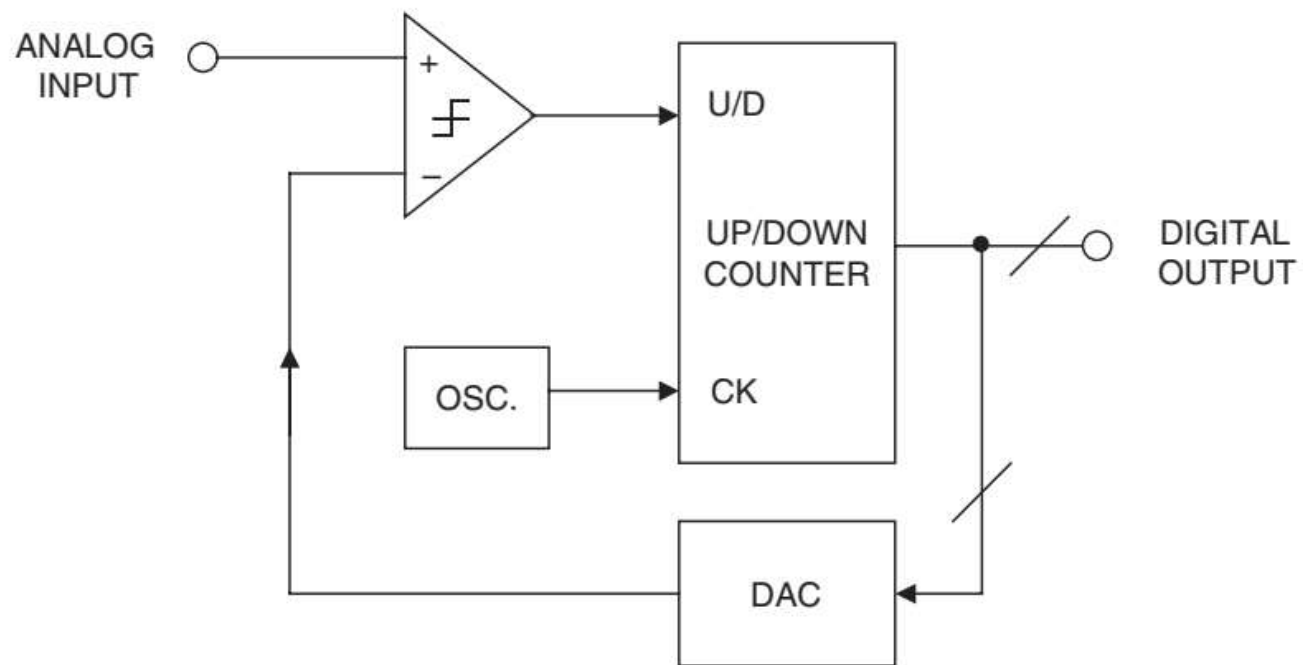
Độ phân giải cao

Giá thành rẻ

Thời gian biến đổi lâu

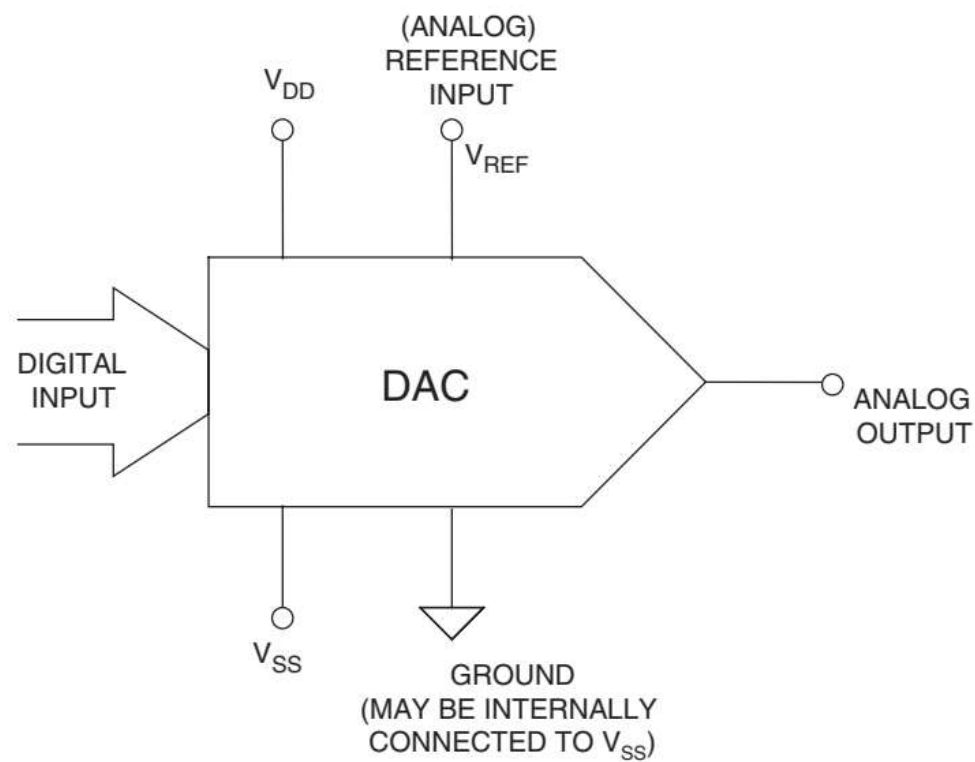
ADC

ADC theo nguyên lý băm



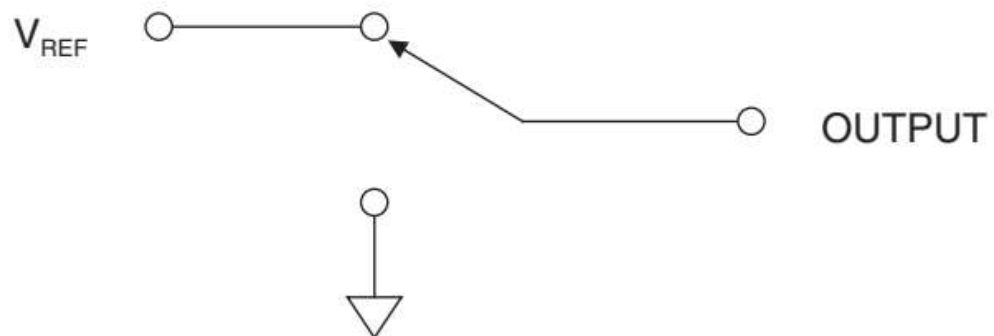
DAC

Cấu trúc DAC



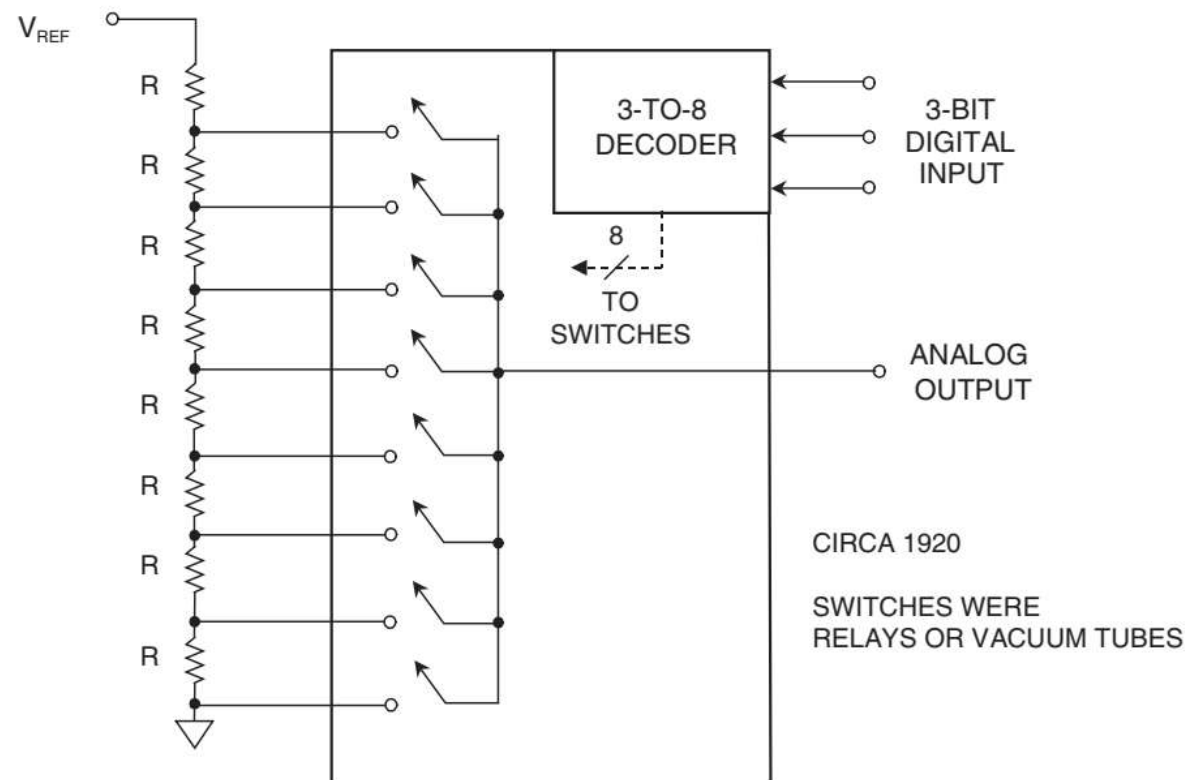
DAC

1 bit DAC:

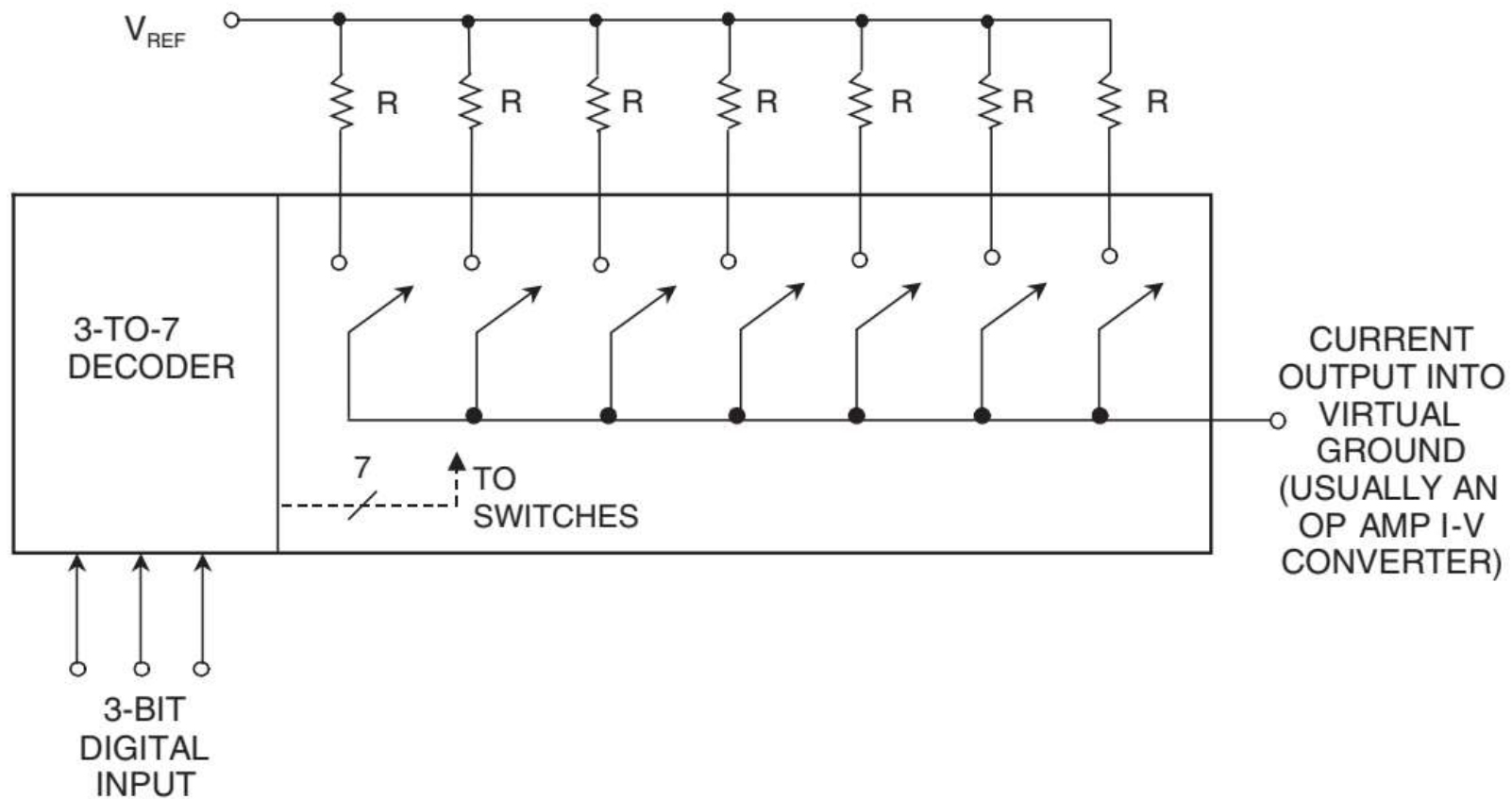


DAC

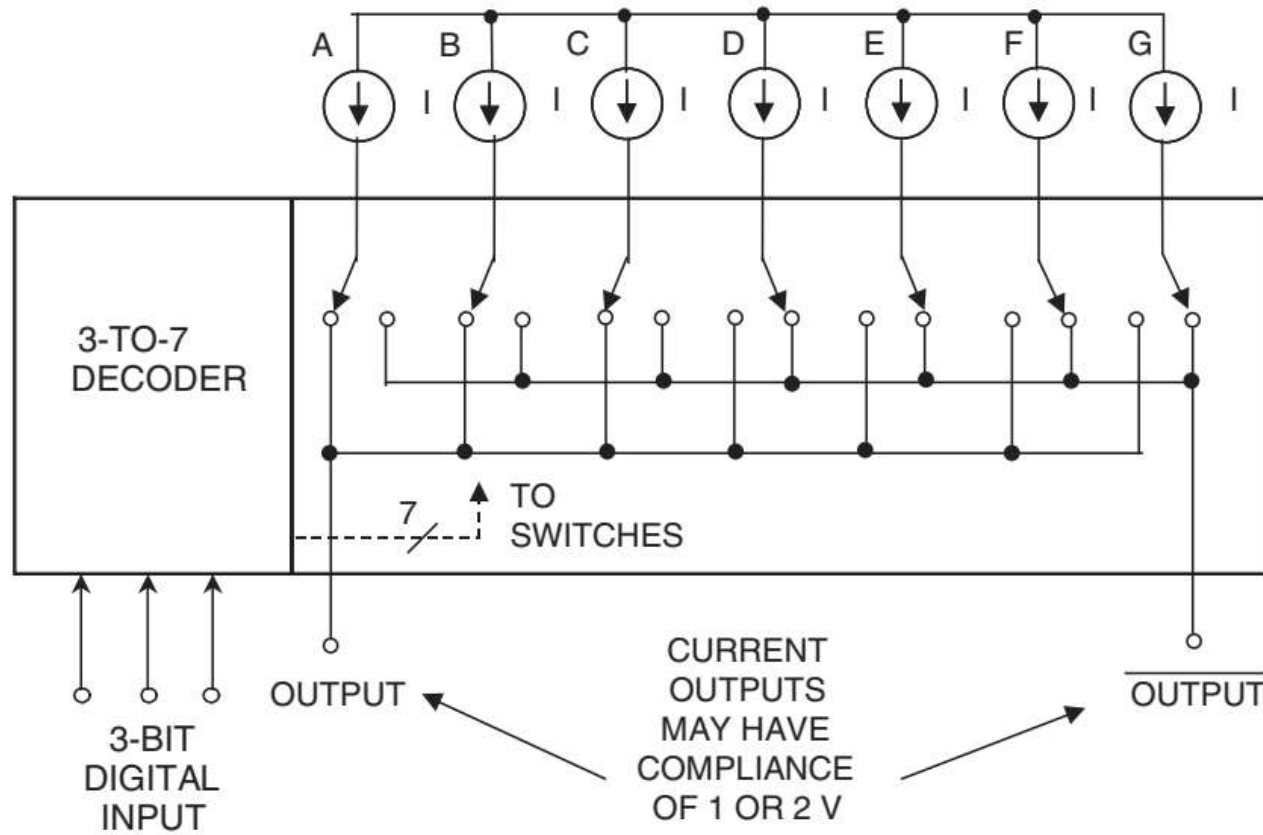
The Kelvin Divider (String DAC)



DAC

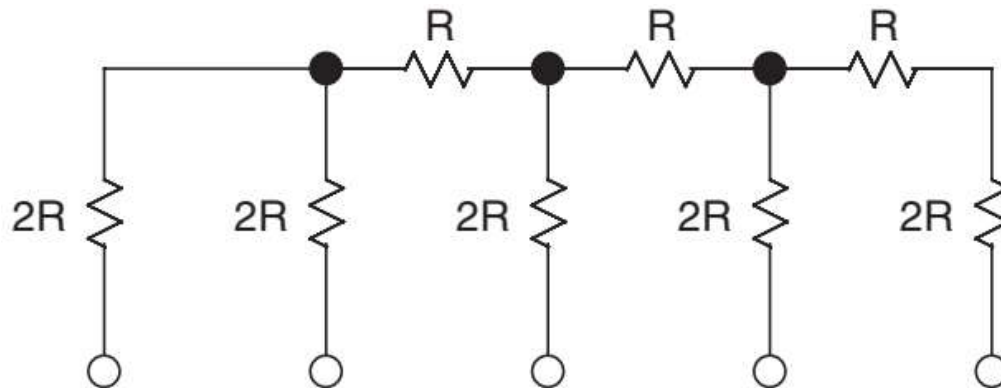


DAC



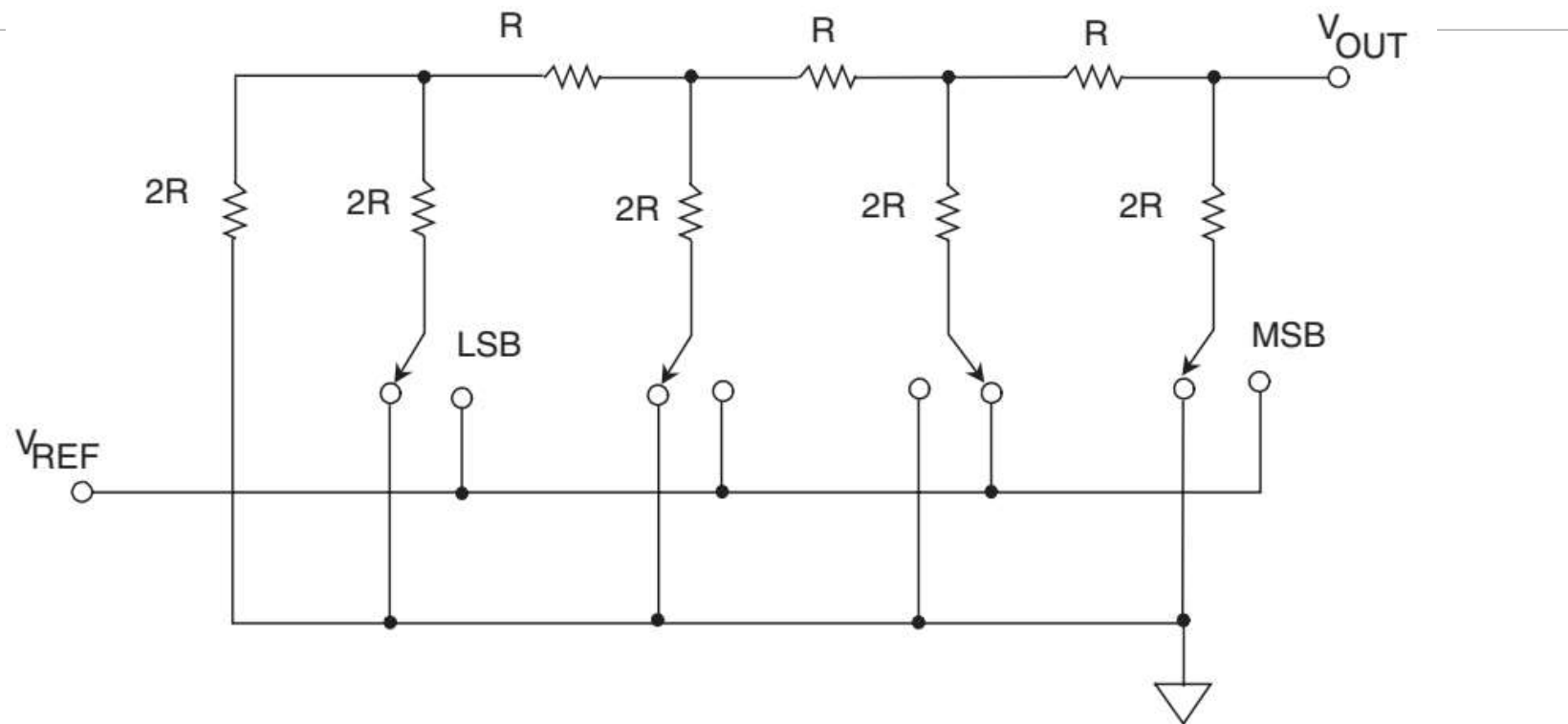
DAC

R-2R DAC: lưới điện trở R-2R



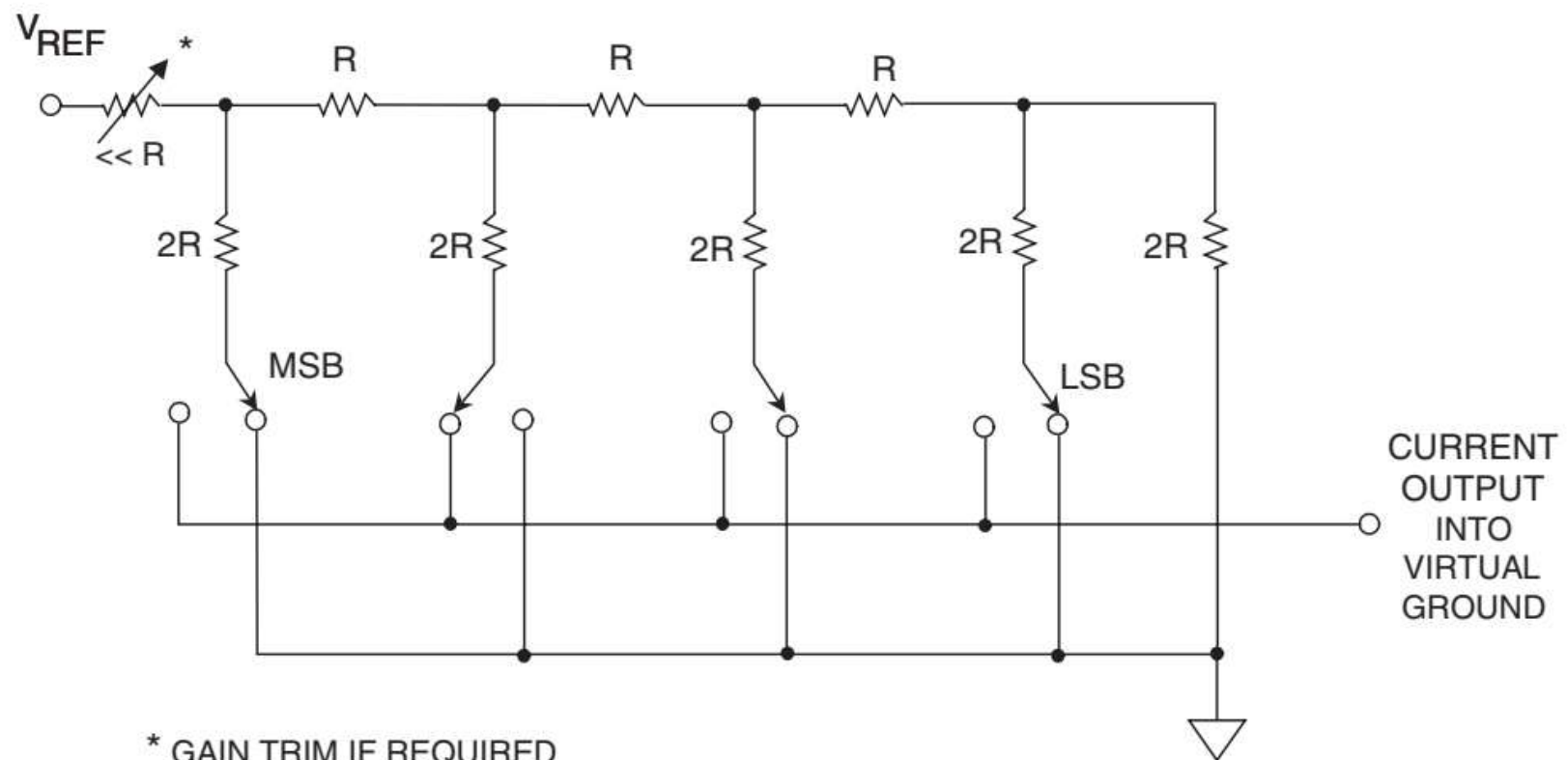
DAC

Volt mode:



DAC

Current Mode:



* GAIN TRIM IF REQUIRED

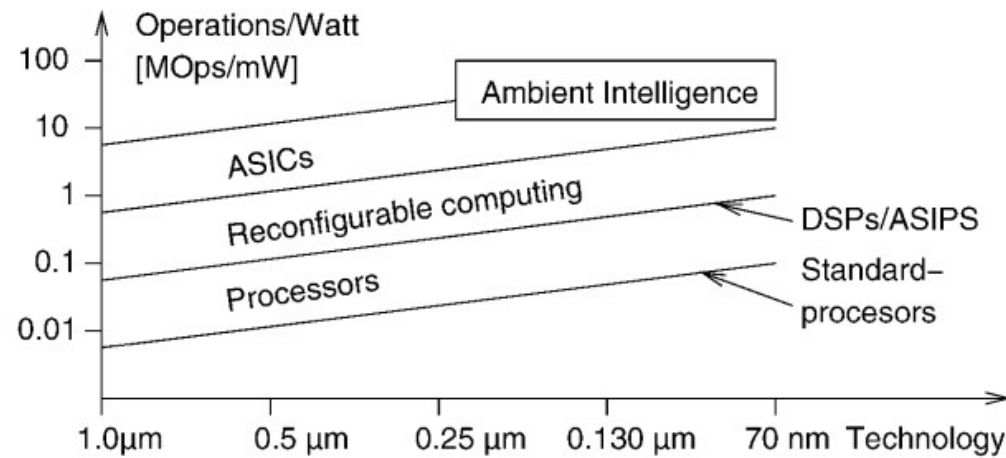
Giao tiếp với Analog

Cơ cấu chấp hành

- Cơ cấu chấp hành có phạm vi lớn từ thiết bị vài trăm tấn đến thiết bị di chuyển um
- Công nghệ vi hệ thống có thể cho thiết bị cực nhỏ để ứng dụng như đưa vào cơ thể con người.
- Các cơ cấu chấp hành khác như thiết bị đóng cắt, truyền động điện, khí nén, thuỷ lực..

CPU

- Để xử lý thông tin ta xem xét đến ASICs (application-specific integrated circuits) sử dụng các mạch cứng, Logic lập trình hay bộ uP lập trình.
- Hiệu quả năng lượng của mỗi công nghệ:



CPU

- Hiệu quả này càng tăng do công nghệ tiến tới kích thước mạch tích hợp ngày càng nhỏ.
- Hiệu quả năng lượng: Mạch cứng > Logic lập trình > uP
- Mức độ linh hoạt: Mạch cứng < Logic lập trình < uP
- Mạch cứng không có độ linh hoạt
- Logic lập trình có thể lập trình nhưng nó giới hạn kích thước của ứng dụng (không đáp ứng ứng dụng lớn và phức tạp)
- uP sử dụng phần mềm có tính linh hoạt cao
- Sử dụng uP được tối ưu hoá cho ứng dụng DSP có thể đạt hiệu suất năng lượng của Logic lập trình
- Với uP tiêu chuẩn hiệu suất năng lượng kém nhất
- Giảm thiểu công suất và năng lượng tiêu thụ đều quan trọng
- Công suất liên qua đến kích thước nguồn, kích thước ổn áp, kích thước kết nối bộ tản nhiệt.
- Giảm thiểu năng lượng đặc biệt quan trọng cho thiết bị di động.

CPU

ASICs

- Đối với các ứng dụng hiệu suất cao và dành cho các thị trường lớn, các mạch tích hợp dành riêng cho ứng dụng (ASIC) có thể được thiết kế.
- Chi phí thiết kế và sản xuất những con chip như vậy khá cao.
- Do đó, ASIC chỉ phù hợp nếu cần hiệu quả sử dụng năng lượng tối đa và nếu thị trường chấp nhận chi phí hoặc nếu có thể bán được một số lượng lớn các hệ thống như vậy.

CPU

Processors

- Ưu điểm chính của bộ xử lý là tính linh hoạt của chúng. Với bộ xử lý, hoạt động chung của hệ thống nhúng có thể được thay đổi chỉ bằng cách thay đổi phần mềm chạy trên bộ xử lý đó
- Có thể chỉnh sửa lỗi và thay đổi bằng phần mềm
- Nó có 2 loại uP và uC. uC là công cụ thực sự cho hệ nhúng
- Thông thường kiến trúc khác với PC

CPU

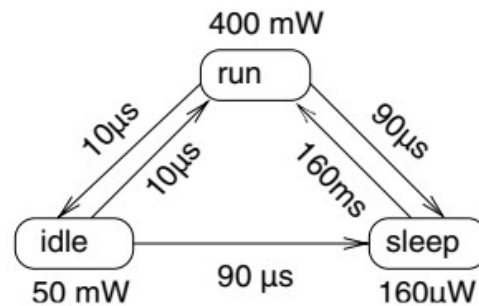
Processors – hiệu quả năng lượng

- Kiến trúc tối ưu để đạt hiệu quả năng lượng và đảm bảo không bị mất trong quá trình tạo ra phần mềm.
- Ví dụ chương trình compiler tạo ra sử dụng 50% số chu kỳ thì hiệu suất kém 50% vì có thể còn phải tăng điện áp và clock để đảm bảo thời hạn.
- Có nhiều cách để tiết kiệm năng lượng và tăng hiệu quả: thiết kế tập lệnh, thiết kế chip..
- Ví dụ: tắt clock của processor trong quá trình nhàn rỗi. Có thể tắt clock cho một số bộ phận chính của chip.

CPU

Processors

- Quản lý công suất động(DMP): Bộ xử lý có một số trạng thái tiết kiệm năng lượng ngoài trạng thái vận hành tiêu chuẩn.
- Mỗi trạng thái có mức tiêu thụ và thời gian chuyển trạng thái khác nhau.
- Idle các ngoại vi vẫn hoạt động. Sleep mọi hoạt động đều tắt



CPU

Processors

- Điều chỉnh điện áp nguồn động
 - Mức tiêu thụ của chip CMOS tăng theo bậc 2 điện áp nguồn

$$P = \alpha C_L V_{dd}^2 f$$

- Trễ của cổng CMOS

$$\tau = k \cdot C_L \cdot \frac{V_{dd}}{(V_{dd} - V_t)^2}$$

CPU

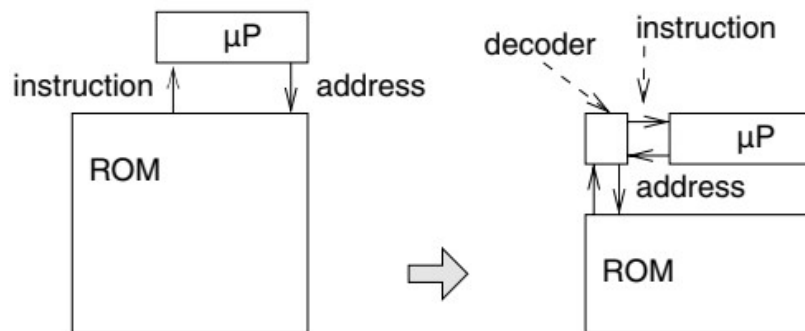
Processors

- Hiệu quả kích thước chương trình: giảm kích thước code rất quan trọng vì ES không có ổ cứng và bộ nhớ hạn chế
- Điều này rõ nhất với SoC, uC vì bộ nhớ tích hợp trên chip
- Bộ nhớ gắn trên chip gọi là bộ nhớ nhúng và có thể chế tạo nó đắt hơn bộ nhớ riêng biệt vì qui trình chế tạo cho VXL và bộ nhớ tương thích.

CPU

Processors

- Các kỹ thuật cải thiện kích thước code:
 - Bộ VXL CISC (Complex Instruction Set Processors): bộ VXL RICS được thiết kế cho tốc độ không phải hiệu quả về kích thước code. CISC đạt hiệu quả về kích thước mã vì hay sử dụng bộ nhớ chậm và ít khi có caches.
 - Nén: Để giảm không gian cũng như năng lượng để tìm và nạp các lệnh thì các lệnh được lưu trữ dưới dạng nén. Nó cũng giảm băng thông nên việc nạp lệnh nhanh hơn.



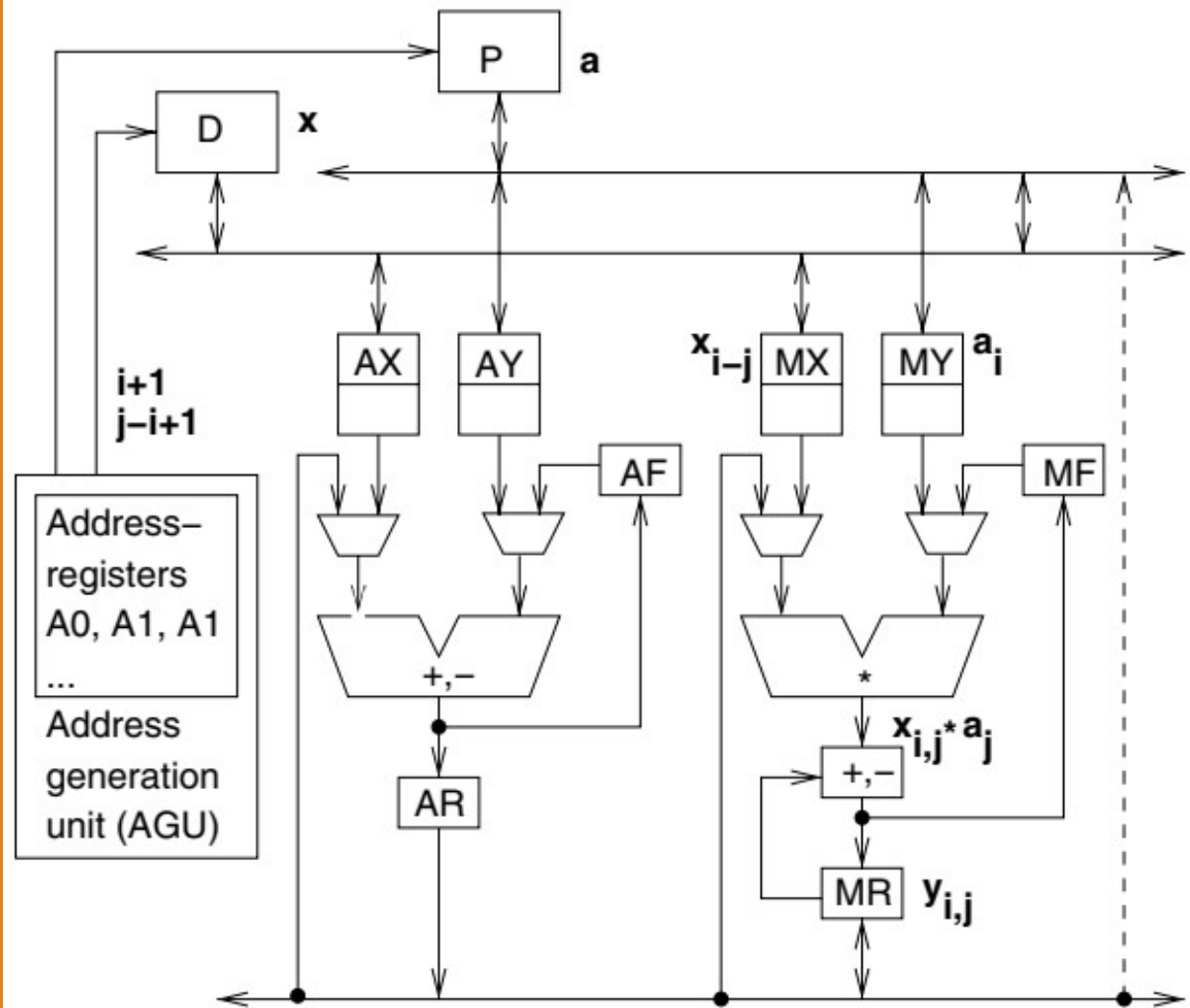
CPU

Processors

- Hiệu quả về thời gian thực thi: Để đáp ứng các hạn chế về thời gian mà không phải sử dụng tần số xung nhịp cao, kiến trúc có thể được tùy chỉnh theo các miền ứng dụng nhất định, chẳng hạn như xử lý tín hiệu số (DSP). Người ta thậm chí có thể tiến thêm một bước nữa và thiết kế bộ xử lý tập lệnh dành riêng cho ứng dụng (ASIP).
- DSP được thiết kế sao cho mỗi lần lặp có thể được mã hóa thành một lệnh đơn. Chúng ta hãy xem xét một ví dụ: Có 2 vùng nhớ D và P, có khối phát địa chỉ AGU, có các thanh ghi AX, AY, AR, MX, MY, MR

CPU

PROCESSORS



CPU

Processors

- uC: một số lớn của VXL cho hệ nhúng là uC. uC không phức tạp và có thể sử dụng dễ dàng.
- uC sử dụng nhiều là 9051 của Intel:
 - 8 bit CPU, optimized for control applications,
 - Large set of operations on Boolean data types,
 - Program address space of 64 k bytes,
 - Separate data address space of 64 k bytes,
 - 4 k bytes of program memory on chip, 128 bytes of data memory on chip,
 - 32 I/O lines, each of which can be addressed individually,
 - 2 counters on the chip,
 - universal asynchronous receiver/transmitter for serial lines available on the chip,
 - clock generation on the chip,
 - many variations commercially available

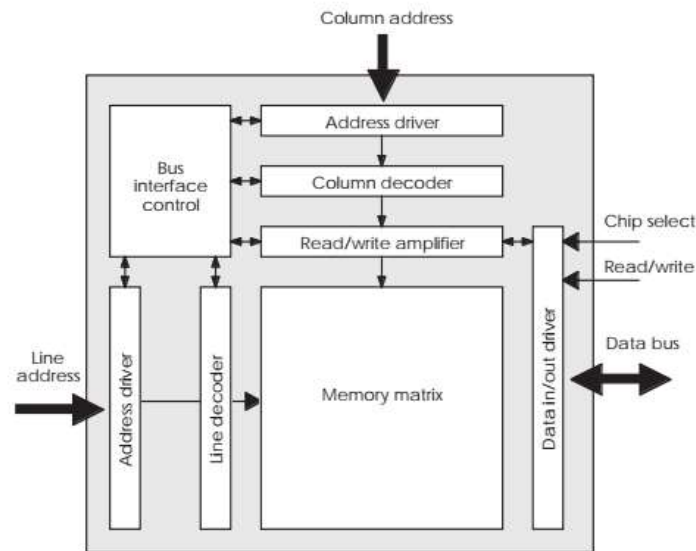
CPU

Logic lập trình

- Trong một số trường hợp chip ASICs quá đắt và uP quá chậm và tiêu thụ nhiều năng lượng thì giải pháp Logic lập trình có thể thực hiện
- Nó nhanh như phần cứng chuyên dụng và có thể lập trình được
- Nó có thể sử dụng trong các lĩnh vực”
 - Tạo các kiểu mẫu nhanh: ASICs thiết kế tốn nhiều thời gian, cần tạo ra nguyên mẫu thử nghiệm gần với hệ thống thiết kế. Nó có thể tiêu thụ điện năng lớn hơn và chậm hơn hệ thống thiết kế.
 - Ứng dụng chế tạo số lượng nhỏ: Nếu thị trường quá nhỏ cho chip ASICs thì Logic lập trình có thể sử dụng. Ví dụ sử dụng FPGA.

Bộ nhớ

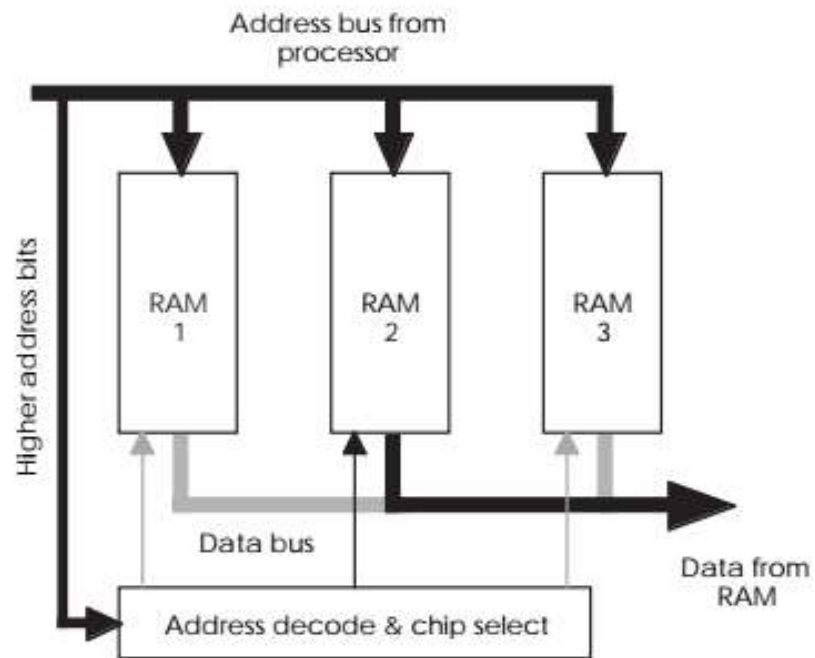
- Dữ liệu, chương trình, cấu hình FPGA được lưu trong một số loại bộ nhớ
- Bộ nhớ có dung lượng theo yêu cầu ứng dụng, hiệu suất, chi phí, năng lượng tiêu thụ theo yêu cầu.
- Bộ nhớ thường không đáp ứng yêu cầu của VXL. Tốc độ bộ nhớ không theo kịp VXL
- Tổ chức bộ nhớ”



Bộ nhớ

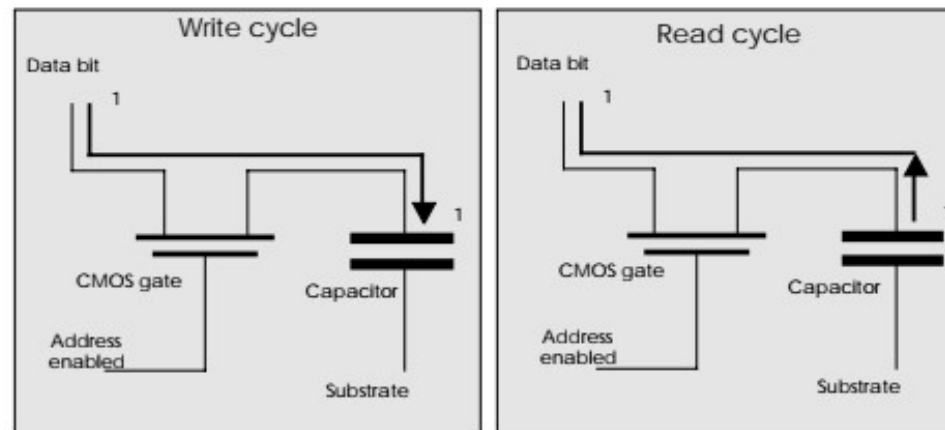
- Các tín hiệu chính của bộ nhớ:

- Bus địa chỉ
- Bus dữ liệu
- Chip selects

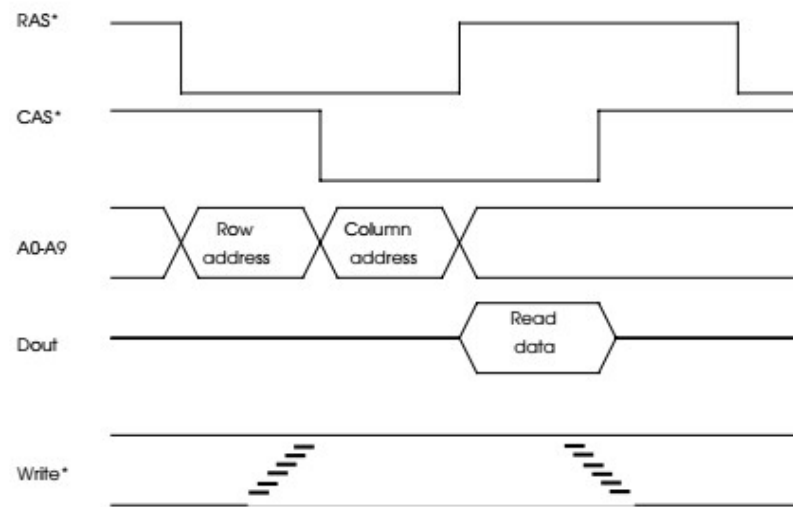


Bộ nhớ

- DRAM:
 - Dùng cho PC và một số thiết bị khác yêu cầu dung lượng rất lớn chi phí thấp
 - Mỗi một bit cần 1 bóng bán dẫn nên đạt mật độ cao giá rẻ, dữ liệu lưu trong 1 tụ nhỏ
 - Cần Refresh thường xuyên. Chi phí cho việc refresh chiếm 3-4% thời gian xử lý có thể chấp nhận được và nó thực hiện bằng bộ ĐK phần cứng



Bộ nhớ



Bộ nhớ

- SRAM
 - Không cần Refresh và giữ nguyên giá trị khi được cấp nguồn, tiêu thụ điện ít và nhanh hơn DRAM
 - Cần nhiều bóng bán dẫn hơn DRAM nên đắt và dung lượng nhỏ hơn.
 - Giao diện đơn giản hơn
 - Sử dụng làm bộ nhớ cache cho VXL nhanh, bộ nhớ chính trong các thiết bị di động, bộ nhớ cho uC
 - Pseudo-static RAM DRAM động nhưng giao tiếp như SRAM
 - Battery backup RAM: SRAM có thể nuôi bằng pin để lưu thông tin khi mất điện

Bộ nhớ

- EPROM và OTP:
 - EPROM lưu chương trình và dữ liệu khi tắt hệ thống.
 - Được ghi và có thể xoá bằng UV chiếu vào cửa sổ
 - Cửa sổ được bảo vệ bằng cách dán nhãn tránh ánh sáng
 - OTP lập trình một lần không xoá được
 - Dữ liệu đảm bảo

Bộ nhớ

- FLASH:
 - Bộ nhớ có thể ghi xoá bằng điện cung cấp thời gian truy cập và mật độ tương tự DRAM
 - Sử dụng 1 bóng cho một bit nhớ, sử dụng điện áp cao để bẫy e- qua lớp oxit cách điện
 - Lập trình mất vài ms
 - Dùng thay thế cho EPROM cho phép lập trình cập nhật từ xa

Bộ nhớ

- EEPROM:
 - Bộ nhớ có thể ghi/xoá bằng điện
 - Chu kỳ ghi/xoá chậm diễn ra trên khối lớn thay vì từng bit/byte
 - Ghi xoá thực hiện bởi điện áp lớn tạo ra bên trong hay cấp từ bên ngoài
 - Khi ghi/xoá thì Transistor bị bào mòn và chỉ có một số lần ghi/xoá nhất định (10k-1M)
 - Không thay thế trực tiếp được cho RAM

Bộ nhớ

- Kiểm tra chẵn/lẻ
 - Việc triển khai ngày càng hạn chế
 - Một bit thêm vào để thực hiện kiểm tra
 - Không phát hiện được hết lỗi
 - Việc thực hiện kiểm tra làm chậm truy cập bộ nhớ do đó giảm hiệu suất xử lý
 - Hệ thống quan trọng cần biết bộ nhớ có lỗi hay không, việc kiểm tra có thể yêu cầu

Bộ nhớ

- Phát hiện và sửa lỗi bộ nhớ: bổ xung thêm các bit trên mỗi byte sử dụng các PP mã hoá để tăng cường tính bảo vệ. Có 2 kiểu thiết kế bộ nhớ:
- Bộ nhớ có phát hiện lỗi: bổ xung thêm vào các bit để phát hiện lỗi nhiều bit. Tùy thuộc số bit bổ xung mà nó có thể phát hiện các ĐK lỗi tốt hơn. Thêm nhiều bit và bộ ĐK phức tạp
- Bộ nhớ có sửa lỗi: nó không chỉ phát hiện mà còn sửa lỗi. Cho phép hệ hoạt động bất chấp lỗi. Đắt tiền nhưng mang lại hiệu quả tốt nhất cho bộ nhớ

Bộ nhớ

- Thời gian truy cập của bộ nhớ cần phù hợp để VXL có thể đọc thông tin chính xác. Nếu bộ nhớ quá chậm thì dữ liệu VXL truy cập sẽ không chính xác.
- Một số VXL có thể truy cập nhiều loại bộ nhớ bằng cách trên trạng thái chờ
- XVL có thể chạy nhanh hơn so với bộ nhớ bên ngoài

Truyền thông

- Thông tin cần truyền trong hệ nhúng
- Đặc biệt trong các ứng dụng IoT
- Các kênh thông tin cần xác định thông số như tốc độ truyền, dung lượng, nhiễu...
- Xác xuất lỗi có thể tính toán theo lý thuyết
- Đường truyền gọi phương tiện truyền thông bao gồm truyền thông không dây và truyền qua cáp (đồng, cáp quang)
- Việc kết nối các hệ thống nhúng không đơn giản
- Có nhiều yêu cầu cho việc truyền thông tin

Truyền thông

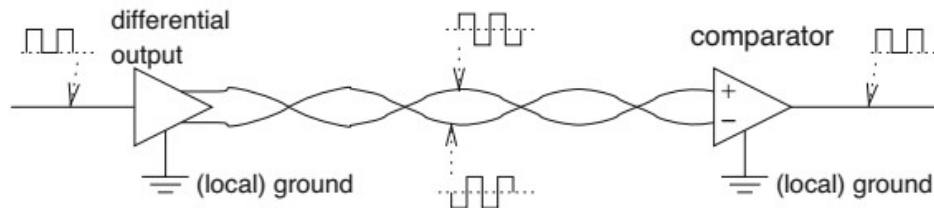
- Yêu cầu của truyền thông mà hệ cần đáp ứng:
 - Hoạt động theo thời gian thực: Yêu cầu cấp thiết đối với việc thiết kế hệ thống truyền thông. Một số giải pháp chi phí thấp như Ethernet tiêu chuẩn không đáp ứng được yêu cầu này.
 - Hiệu quả: Việc kết nối các thành phần phần cứng khác nhau có thể tốn kém. Ví dụ, kết nối điểm-điểm trong các tòa nhà lớn gần như không thể thực hiện được. Với các dây riêng biệt, việc thêm các thành phần mới cũng khó khăn. Nhu cầu về hiệu quả chi phí cũng ảnh hưởng đến cách thức cung cấp năng lượng cho các thiết bị bên ngoài. Thường xuyên có nhu cầu sử dụng nguồn điện trung tâm để giảm chi phí.
 - Băng thông và độ trễ truyền thông phù hợp: Yêu cầu về băng thông của hệ thống nhúng có thể khác nhau. Điều quan trọng là cung cấp đủ băng thông mà không làm cho hệ thống truyền thông trở nên quá đắt đỏ.

Truyền thông

- Hỗ trợ giao tiếp theo hướng sự kiện: Các hệ thống dựa trên polling cung cấp hành vi theo thời gian thực rất có thể dự đoán được. Tuy nhiên, độ trễ liên lạc của họ có thể quá lớn và cần có cơ chế để liên lạc nhanh, hướng tới sự kiện.
- Bảo mật/quyền riêng tư: Việc đảm bảo tính bảo mật/quyền riêng tư của thông tin bí mật (bảo mật) có thể yêu cầu sử dụng mã hóa.
- An toàn/mạnh mẽ: Đối với các hệ thống quan trọng về an toàn, phải đạt được mức độ an toàn cần thiết. Điều này bao gồm tính bền vững: các hệ thống vật lý không gian mạng có thể được sử dụng ở nhiệt độ khắc nghiệt, gần các nguồn bức xạ điện từ chính, v.v.
- Khả năng chịu lỗi: Bất chấp mọi nỗ lực để đảm bảo độ bền, lỗi vẫn có thể xảy ra. Các hệ thống vật lý điện tử phải hoạt động ngay cả sau khi xảy ra lỗi, nếu khả thi. Việc khởi động lại, giống như những lần khởi động lại trong PC, không thể được chấp nhận.
- Khả năng bảo trì, chẩn đoán: Rõ ràng, có thể sửa chữa các hệ thống nhúng trong khung thời gian hợp lý.

Truyền thông

- Bền vững với nhiễu: sử dụng hệ thống truyền vi sai



Năng lượng

- Một trong mục tiêu ưu tiên khi thiết kế ES là hiệu quả năng lượng
- Nguồn năng lượng: Với thiết bị dung điện lưới năng lượng luôn có sẵn. Với các thiết bị khác năng lượng cấp bằng các cách khác nhau. Đặc biệt các thiết bị IoT năng lượng rất ít. Pin sử dụng năng lượng hoá học. Việc cấp pin cho thiết bị khó khăn. Ta sử dụng một số biện pháp thu hoạch năng lượng:
 - Sử dụng pin mặt trời
 - Sử dụng gốm áp điện. Cơ năng \rightarrow điện năng
 - Sử dụng pin nhiệt biến sự chênh nhiệt thành điện
 - Biến đổi động năng thành điện
 - Thu hoạch bức xạ điện từ xung quanh
 - Một số tác động vật lý khác...

Năng lượng

- Thiết bị lưu trữ năng lượng
 - Pin không sạc
 - Tụ điện, siêu tụ
 - Pin sạc được

Bảo mật phần cứng

- Yêu cầu chung cho hệ thống nhúng thường có bảo mật
- Quan trọng với IoT
- Phát triển phần cứng bảo mật đặc biệt
- Bảo mật đảm bảo cho truyền thông và lưu trữ
- Bảo mật được thực hiện bất chấp các cuộc tấn công các biện pháp đối phó phải được thiết kế
- Có một số loại tấn công sau:

Bảo mật phần cứng

- Tấn công phần mềm dựa trên việc thực thi phần mềm. Triển khai trojan là ví dụ. Các lỗi phần mềm có thể được khai thác. Tràn bộ nhớ đệm là lỗi gây ra mối nguy hiểm bảo mật. Tấn công vào kênh truyền thông khai thác các thông tin bổ xung trong kênh.
- Tấn công yêu cầu quyền truy cập vật lý
 - Cố gắng mở thêm kênh bằng cách can thiệp vào phần vật lý. Mở chip can thiệp vi mô soi chiếu quang học. Khó nhưng tiết lộ nhiều chi tiết về chip
 - Phân tích công suất. Các PP phân tích có thể chỉ ra cách mã khoá. Sử dụng thống kê để nâng cao độ chính xác
 - Phân tích bức xạ điện từ.