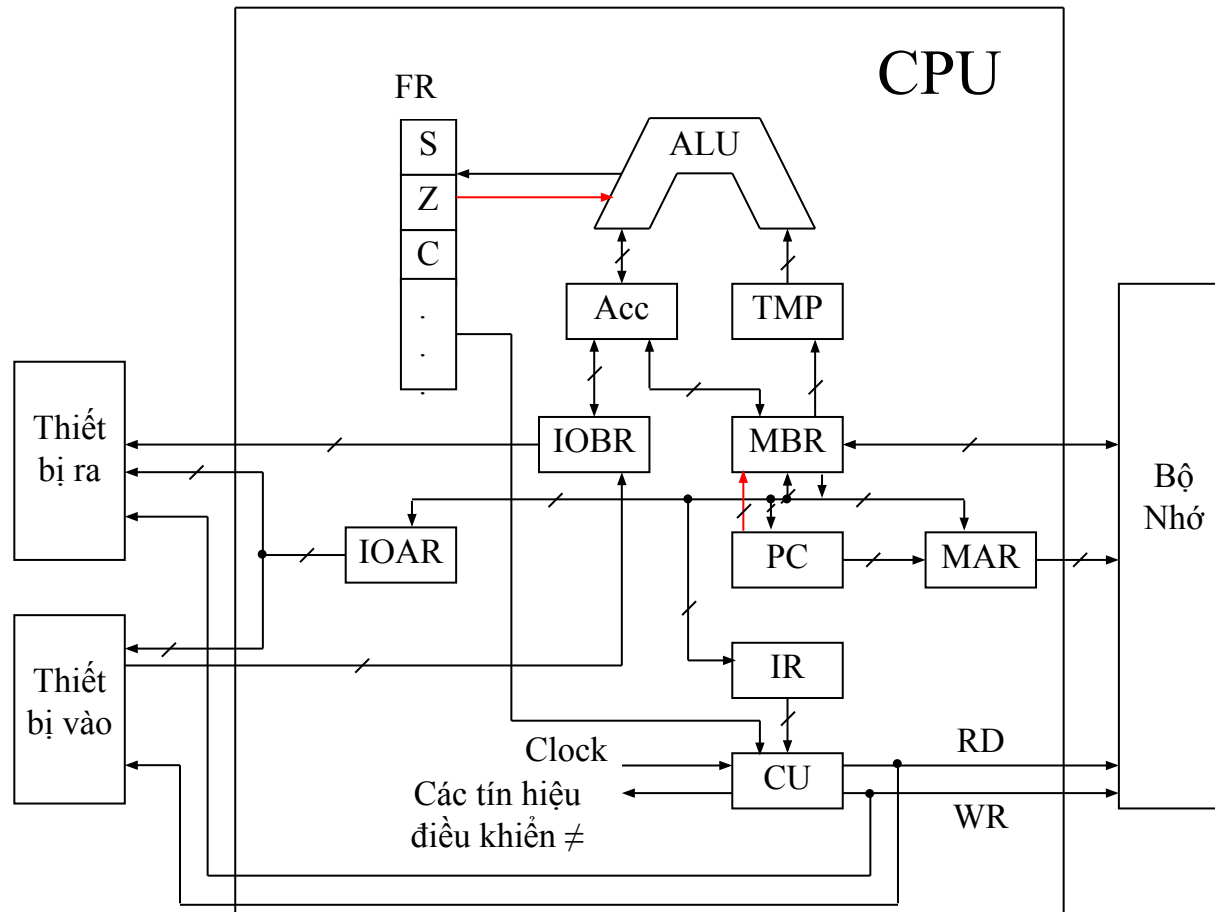
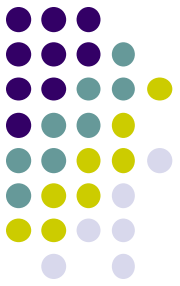


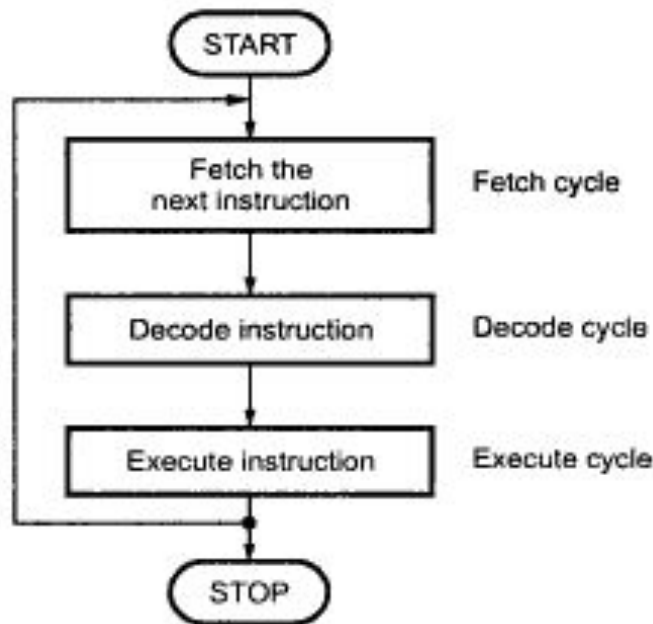
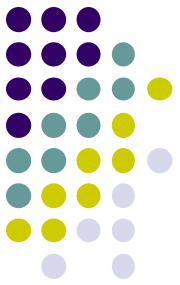
CPU – Bus – IOS

- 3.1. Kiến trúc cơ bản của một máy tính điện tử số và đơn vị xử lý trung tâm (Central Processing Unit - CPU).
- 3.2. Đường truyền
- 3.3. Hệ thống vào/ra

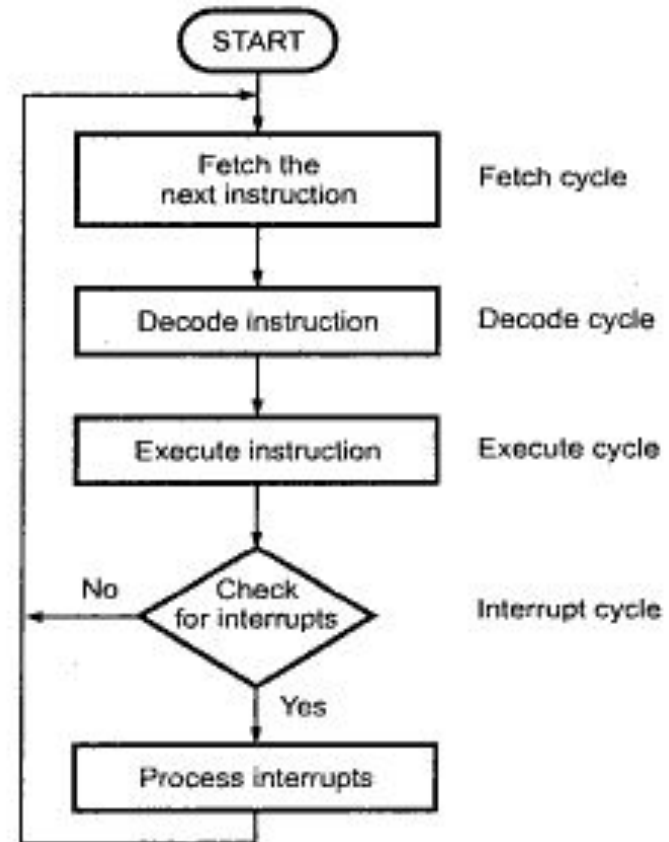
Kiến trúc cơ bản của máy tính điện tử số



Hoạt động của máy tính



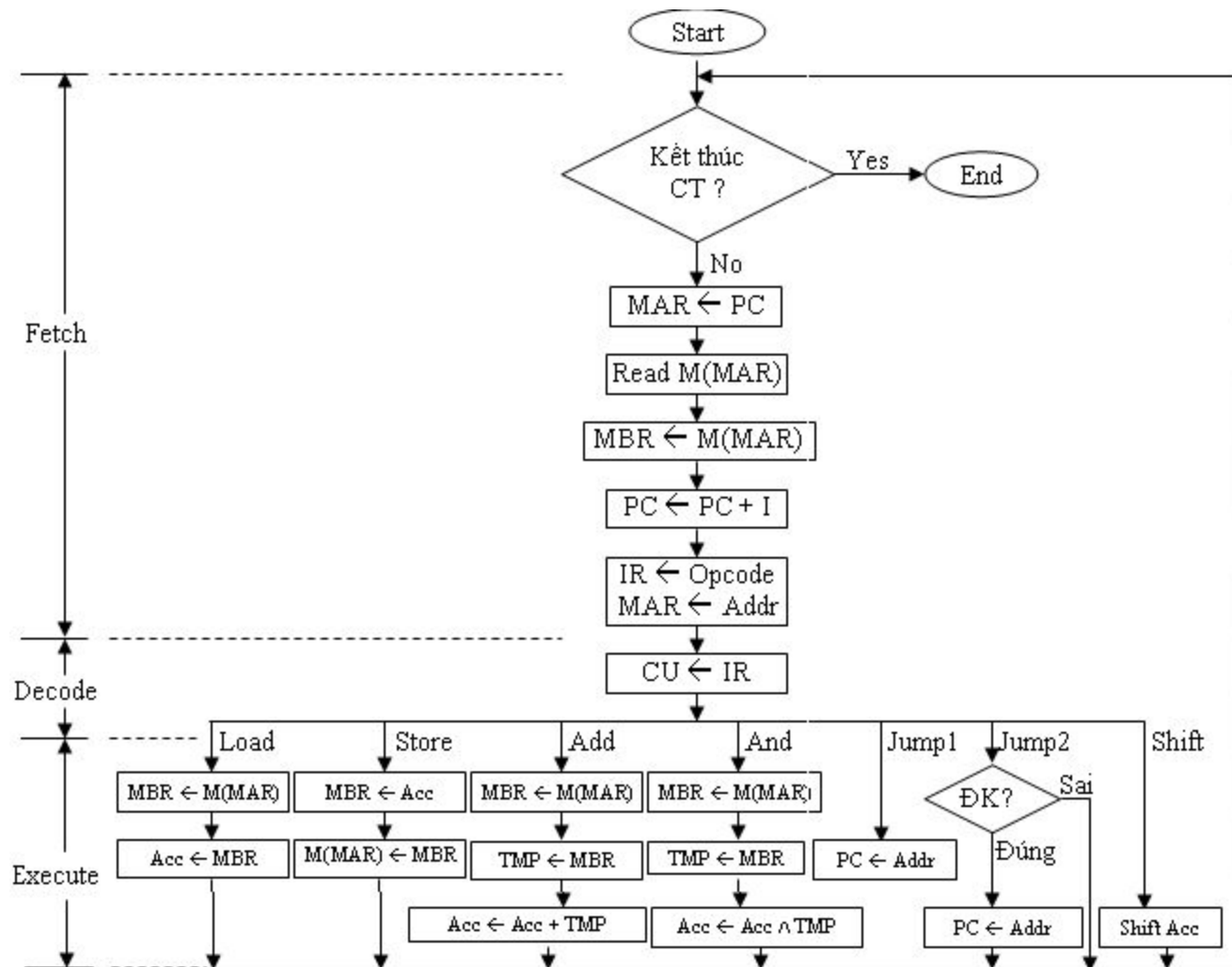
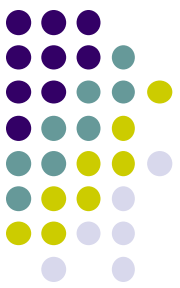
a)



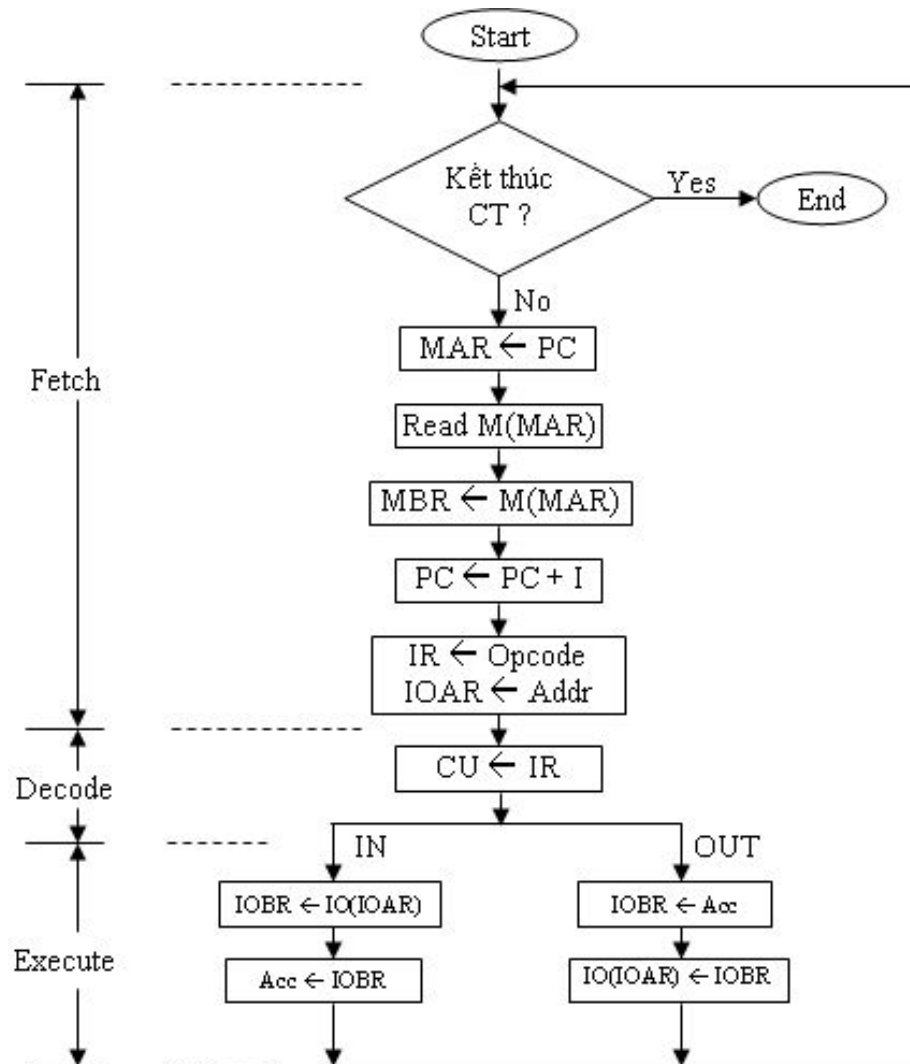
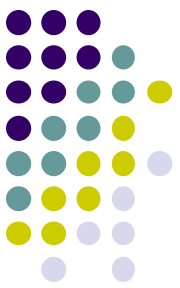
b)

a) Chu kỳ lệnh cơ bản; b) Chu kỳ lệnh cơ bản với ngắt

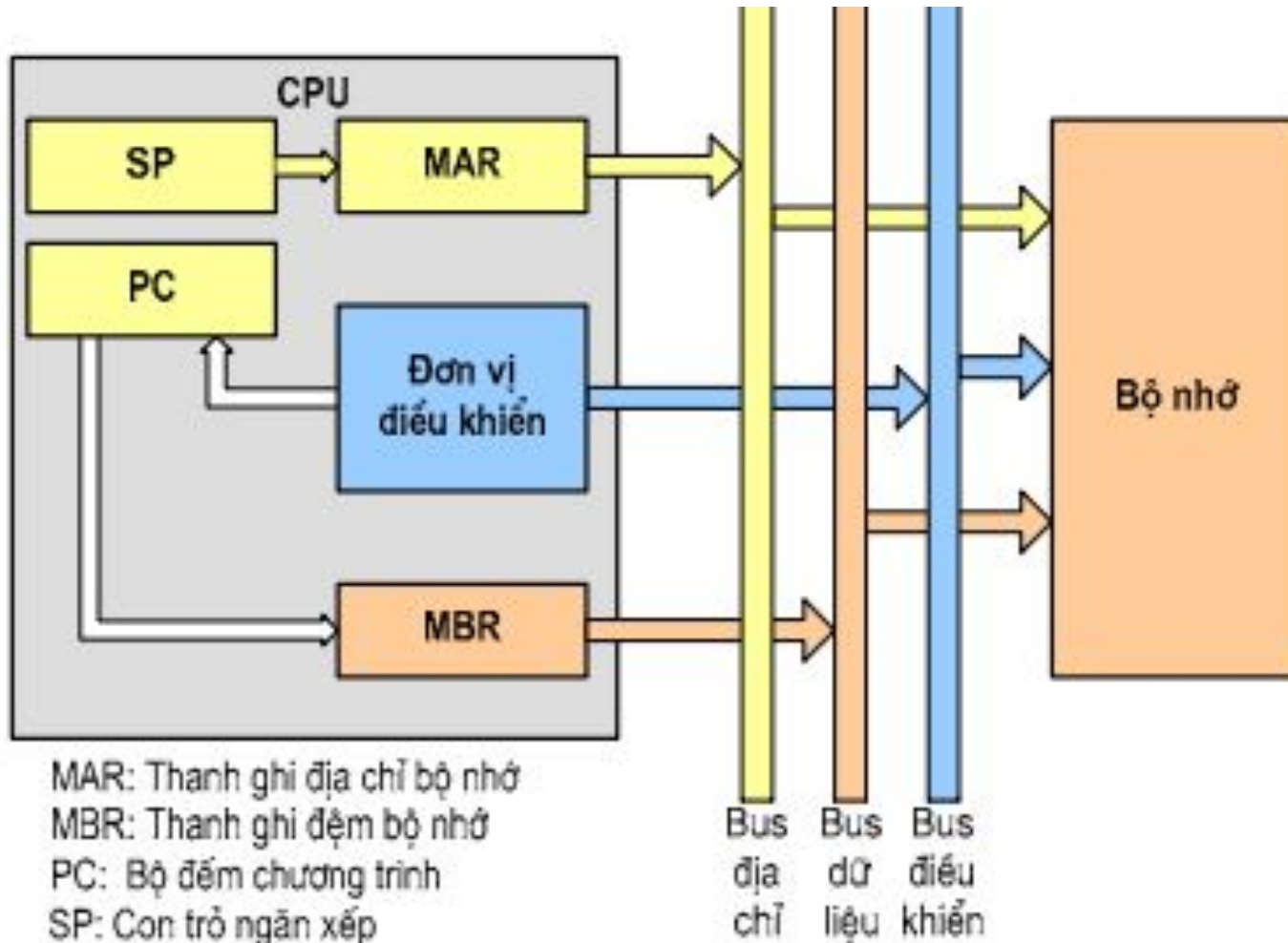
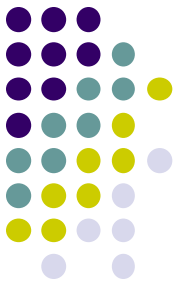
Các thao tác cơ bản khi thực hiện lệnh thao tác với bộ nhớ



Thao tác cơ bản khi thực hiện lệnh vào/ra dữ liệu trực tiếp với thiết bị vào/ra



Chu kỳ lệnh với ngắt

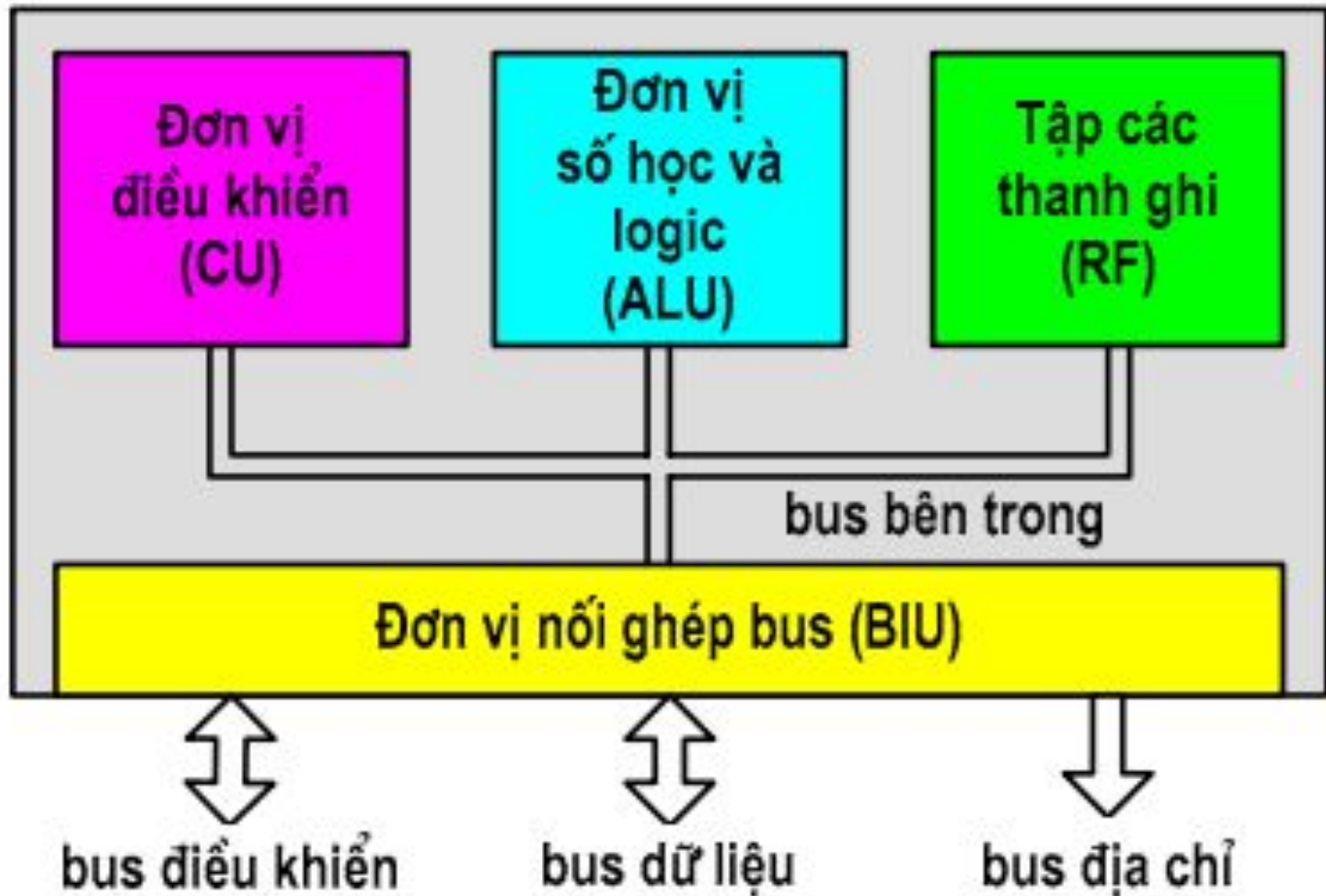
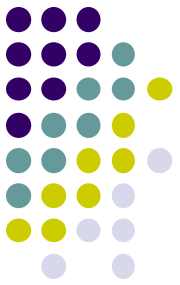




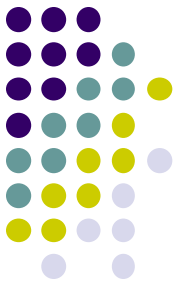
Hoạt động của chu trình ngắt

- Nội dung của bộ đếm chương trình PC (địa chỉ trở về sau khi ngắt) được đưa ra bus dữ liệu.
- ☐ CPU đưa địa chỉ (thường được lấy từ con trỏ ngăn xếp SP) ra bus địa chỉ.
- ☐ CPU phát tín hiệu điều khiển ghi bộ nhớ
- ☐ Địa chỉ trở về trên bus dữ liệu được ghi ra vị trí xác định (ở ngăn xếp).
- ☐ Địa chỉ lệnh đầu tiên của chương trình con điều khiển ngắt được nạp vào PC.

Sơ đồ cấu trúc cơ bản của CPU



Các thành phần cơ bản của CPU



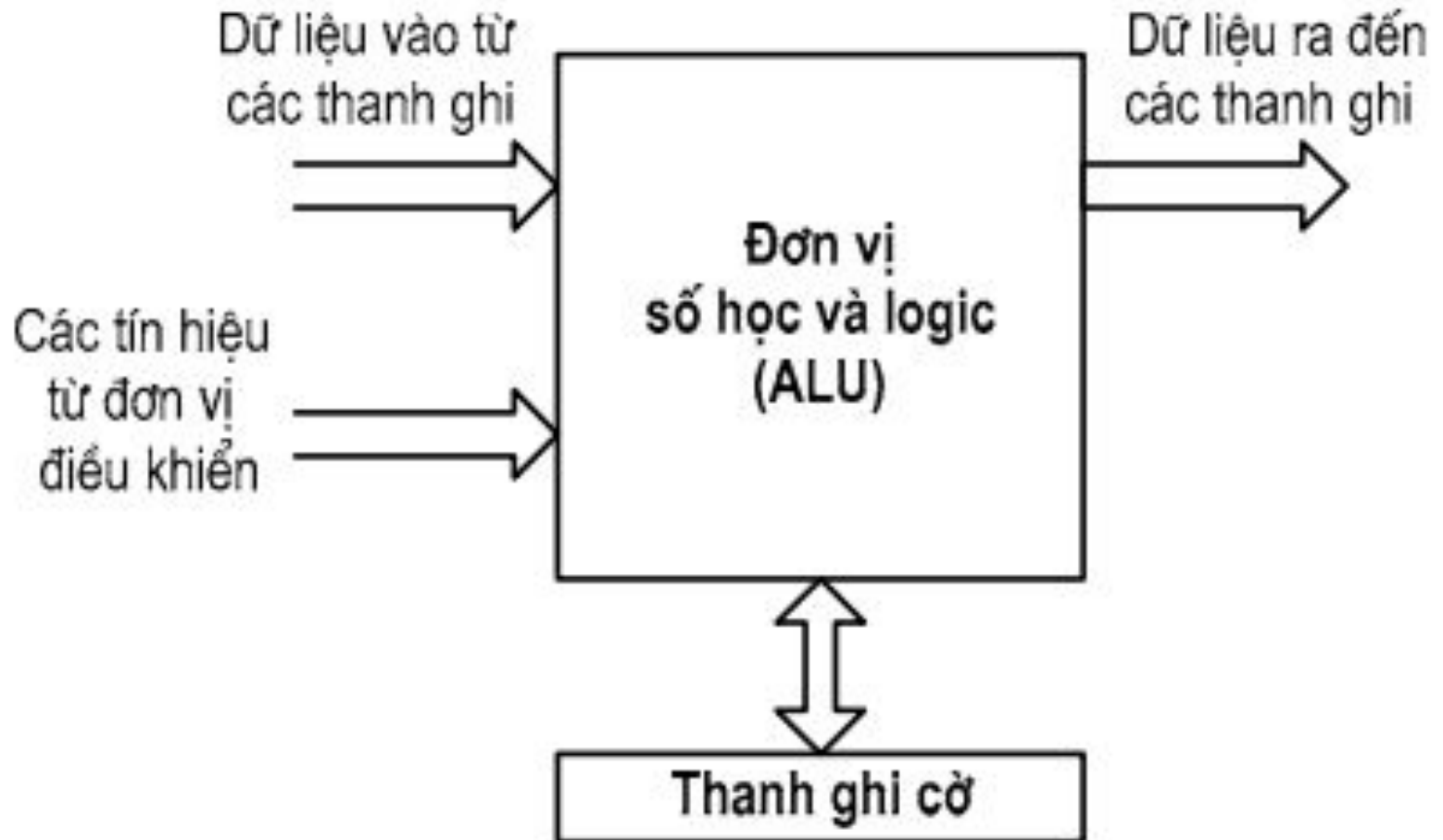
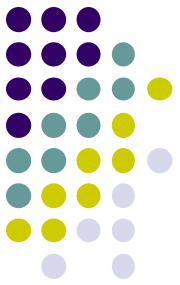
- Đơn vị điều khiển (Control Unit - CU).
- ☐ Đơn vị số học và logic (Arithmetic and Logic Unit - ALU).
- ☐ Tập thanh ghi (Register File - RF).
- ☐ Đơn vị nối ghép bus (Bus Interface Unit - BIU).
- Bus bên trong (Internal Bus).

Đơn vị số học và logic (ALU)

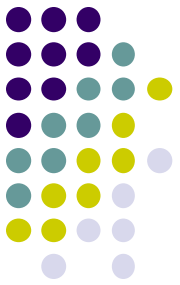


- Chức năng: Thực hiện các phép toán số học và phép toán logic:
 - Số học: cộng, trừ, nhân, chia, tăng, giảm, đảo dấu.
 - Logic: AND, OR, XOR, NOT, phép dịch bit.

Mô hình kết nối ALU



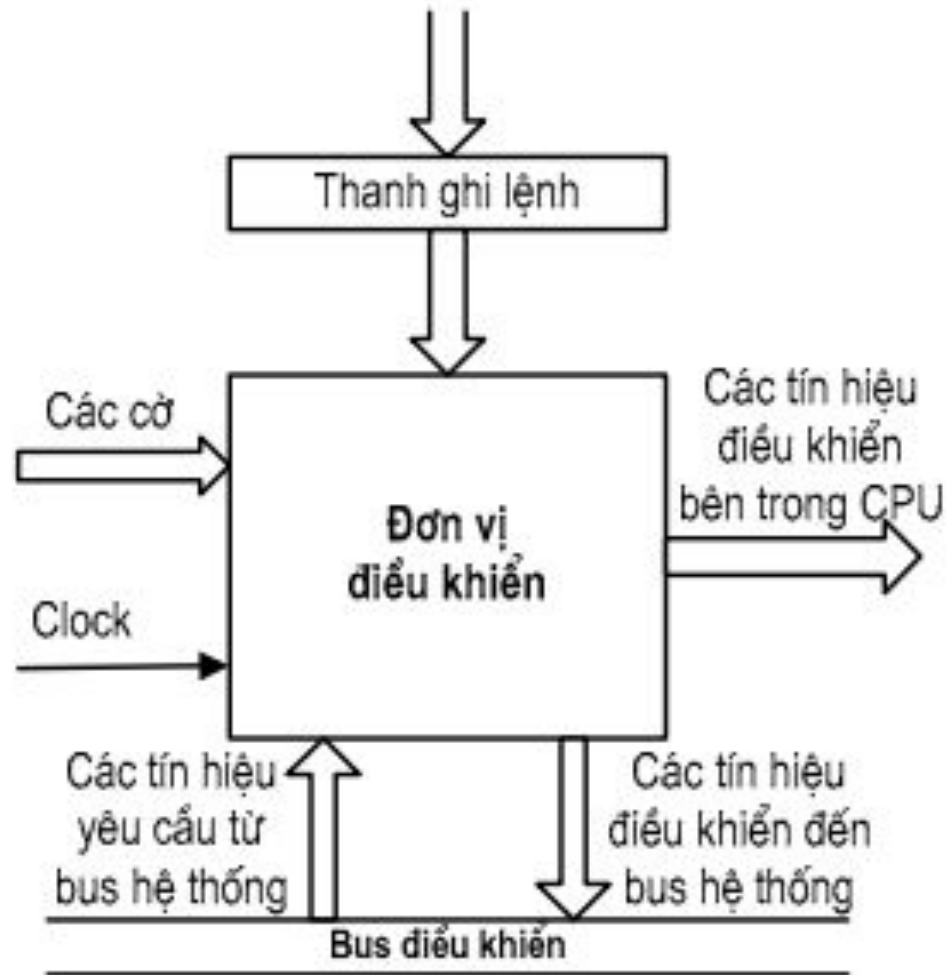
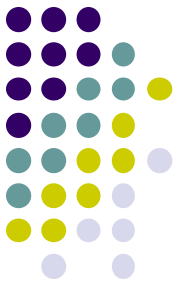
Đơn vị điều khiển (CU:Control Unit)



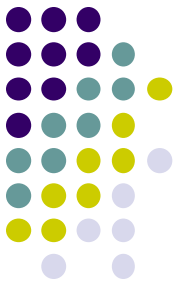
- Chức năng

- ☐ Điều khiển nhận lệnh từ bộ nhớ đưa vào thanh ghi lệnh
- ☐ Tăng nội dung của PC để trở sang lệnh kế tiếp
- ☐ Giải mã lệnh đã được nhận để xác định thao tác mà lệnh yêu cầu
- ☐ Phát ra các tín hiệu điều khiển thực hiện lệnh
- ☐ Nhận các tín hiệu yêu cầu từ bus hệ thống và đáp ứng với các yêu cầu đó.

Mô hình kết nối đơn vị điều khiển

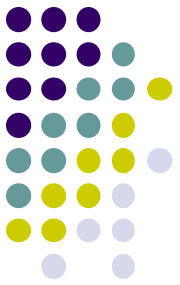


Các tín hiệu đưa đến đơn vị điều khiển



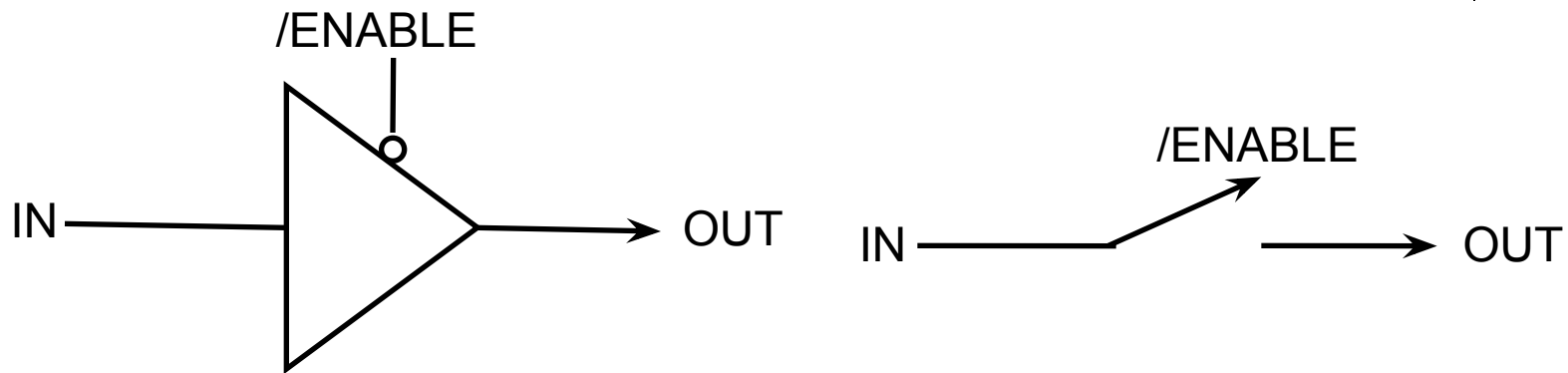
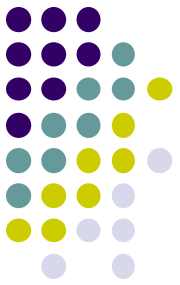
- Clock: tín hiệu nhịp từ mạch tạo dao động bên ngoài.
- ☐ Mã lệnh từ thanh ghi lệnh đưa đến để giải mã.
- ☐ Các cờ từ thanh ghi cờ cho biết trạng thái của CPU.
- ☐ Các tín hiệu yêu cầu từ bus điều khiển

Các tín hiệu phát ra từ đơn vị điều khiển



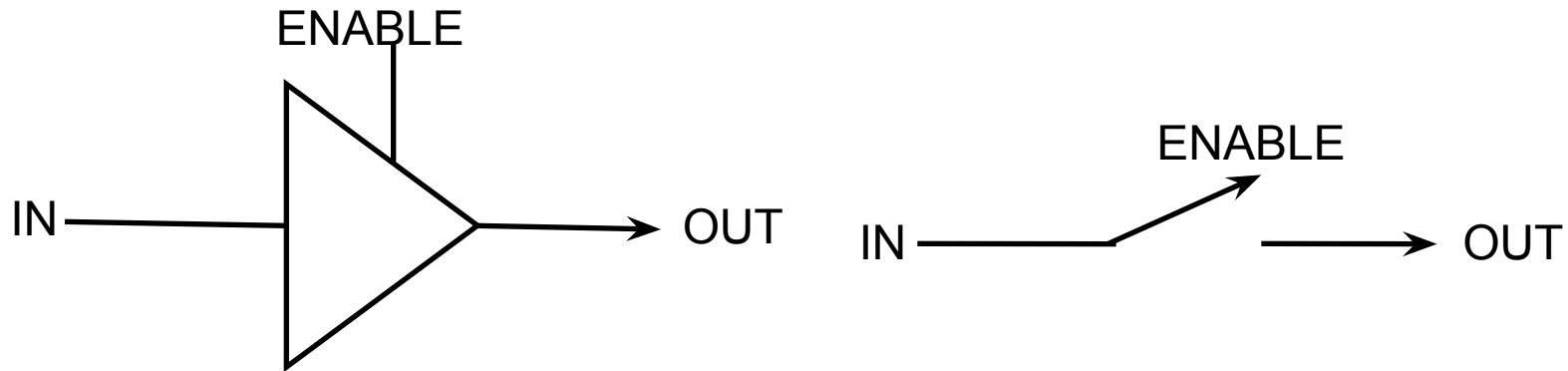
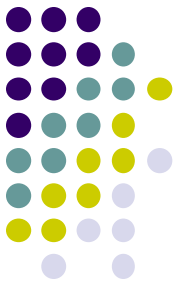
- Các tín hiệu điều khiển bên trong CPU:
 - □ Điều khiển các thanh ghi
 - □ Điều khiển ALU
- Các tín hiệu điều khiển bên ngoài CPU:
 - Điều khiển bộ nhớ
 - Điều khiển các mô-đun vào-ra

Thiết bị 3 trạng thái



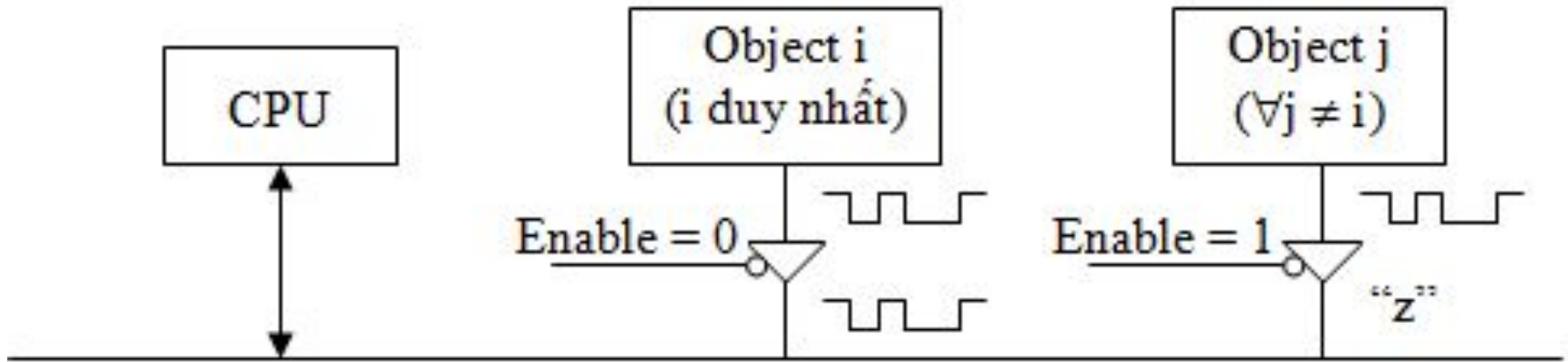
- Hoạt động:
 - Khi chân /ENABLE có mức logic thấp (0) thì đầu ra (OUT) bằng đầu vào (IN).
 - Khi chân /ENABLE có mức logic cao (1) thì đầu ra và đầu vào cách ly (đầu ra có mức trở kháng cao).

Thiết bị 3 trạng thái



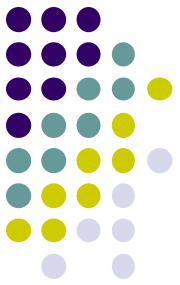
- Hoạt động:
 - Khi chân ENABLE có mức logic cao (1) thì đầu ra (OUT) bằng đầu vào (IN).
 - Khi chân ENABLE có mức logic thấp (0) thì đầu ra và đầu vào cách ly (đầu ra có mức trở kháng cao).

Ghép nối thiết bị với bus

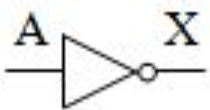


Sơ đồ điều khiển kết nối bus

Các cổng logic cơ bản

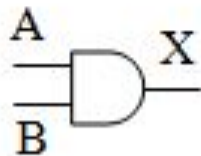


NOT



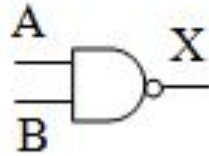
In	out
A	X
0	1
1	0

AND



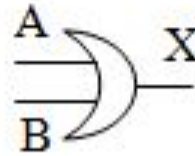
in		out
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

NAND



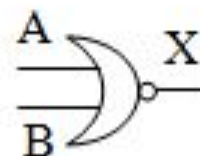
in		out
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

OR



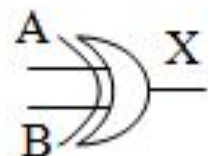
in		out
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

NOR



in		out
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

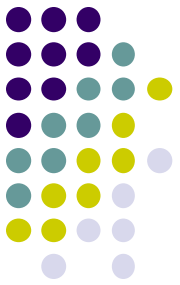
XOR



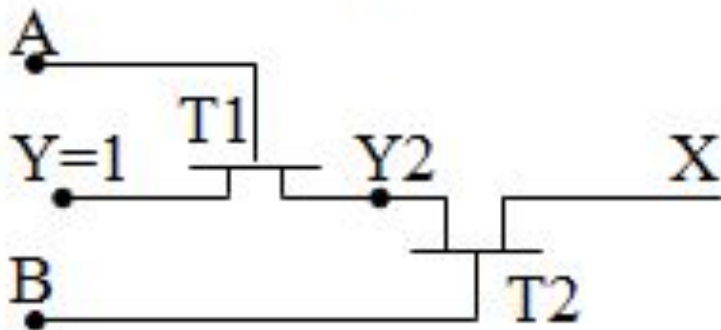
in		out
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Ký hiệu và bảng chân lý của các cổng logic cơ bản

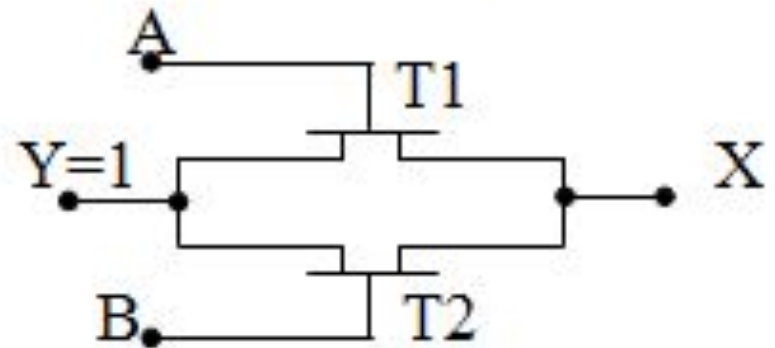
Xây dựng cổng AND và OR từ transistor trường



AND



OR

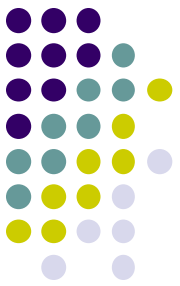


T1

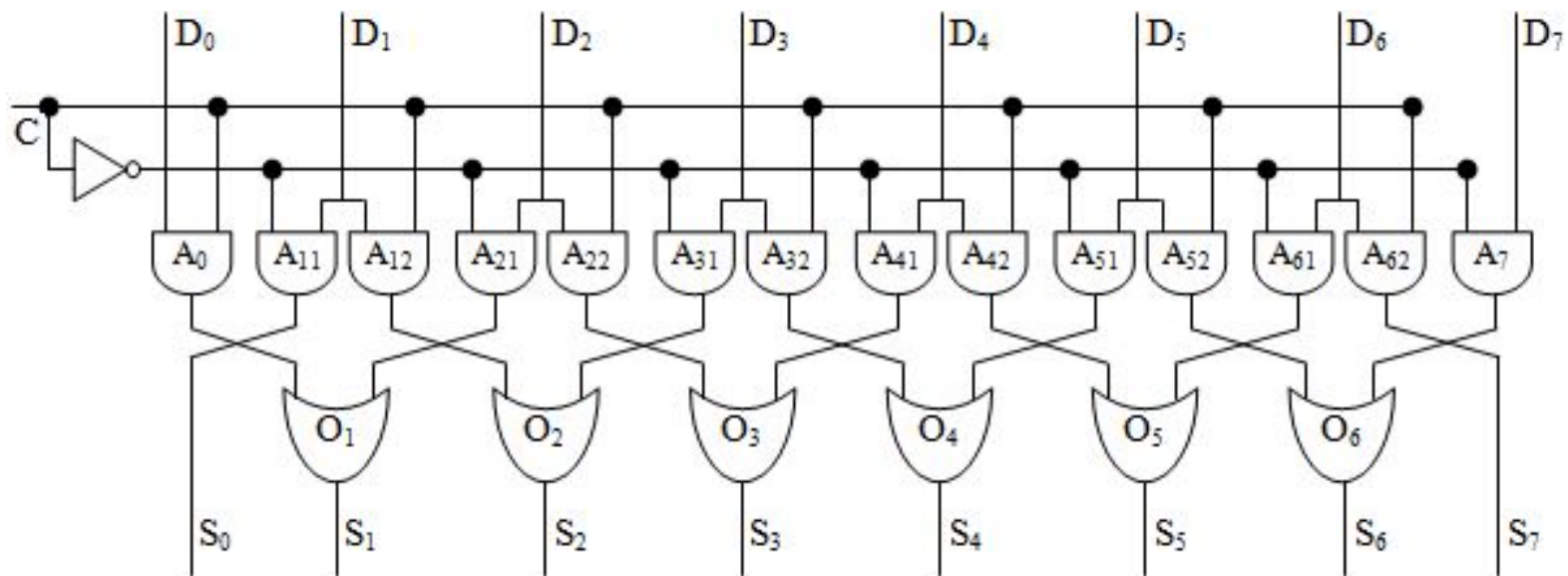
T2

A = 1 thì Công tắc T1 đóng
B = 1 thì công tắc T2 đóng

Một số mạch chức năng của ALU

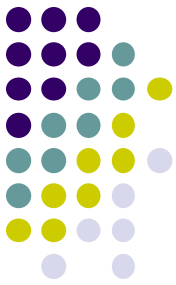


- Bộ ghi dịch được xây dựng từ các cổng logic cơ bản.



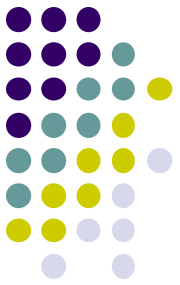
Trong đó: $D_7 \div D_0$: dữ liệu đầu vào
 $S_7 \div S_0$: dữ liệu đầu ra

C: bit điều khiển hướng dịch

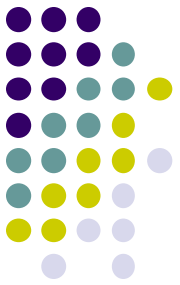


D7	D6	D5	D4	D3	D2	D1	D0
D6	D5	D4	D3	D2	D1	D0	0
S7	S6	S5	S4	S3	S2	S1	S0

D7	D6	D5	D4	D3	D2	D1	D0
0	D7	D6	D5	D4	D3	D2	D1
S7	S6	S5	S4	S3	S2	S1	S0



- Xét $C = 0 \Rightarrow !C = 1$
- $S_0 = \text{AND}(!C, D_1) = \text{AND}(1, D_1) = D_1$
- $S_1 = \text{OR}(A_0, A_2)$
- $A_0 = \text{AND}(C, D_0) = \text{AND}(0, D_0) = 0$
- $A_2 = \text{AND}(!C, D_2) = \text{AND}(1, D_2) = D_2$
- $S_1 = \text{OR}(0, D_2) = D_2$
- $S_7 = A_6 = \text{AND}(C, D_6) = \text{AND}(0, D_6) = 0$
- $\square C=0 \square$ Dịch phải.
- Chứng minh $C = 1 \square$ Dịch trái



Minh họa dịch trái và phải

- Dịch trái và dịch phải

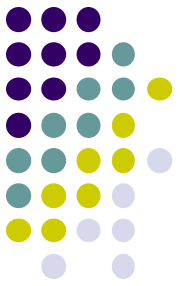


Vào: $D_3D_2D_1D_0$ Ra: $D_2D_1D_00$

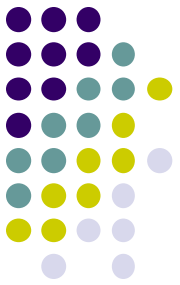


Vào: $D_3D_2D_1D_0$ Ra: $0D_3D_2D_1$

Dịch trái là nhân với 2, dịch phải chia cho 2



- $2 = 10b$ dịch trái 1 bit: $100b = 4$, $1000b = 8$, $10000b = 16$, ... Dịch trái một số n bit thì tương đương nhân số đó với 2^n .
- $1000b = 8$, dịch phải 1 bit: $100b = 4$, dịch phải 1 bit: $10b = 2$ □ Nếu dịch phải 1 số đi n bit thì tương đương chia số đó cho 2^n .



Minh họa với $C = 0$

- Dịch trái:

- $S7 = D6$

D7	D6	D5	D4	D3	D2	D1	D0
D6	D5	D4	D3	D2	D1	D0	0
S7	S6	S5	S4	S3	S2	S1	S0

- $S6 = D5$

- $S5 = D4$

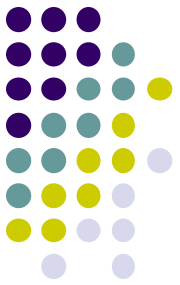
- $\dots S1 = D0, S0 = 0$

D7	D6	D5	D4	D3	D2	D1	D0
0	D7	D6	D5	D4	D3	D2	D1
S7	S6	S5	S4	S3	S2	S1	S0

- Dịch phải:

- $S7 = 0$

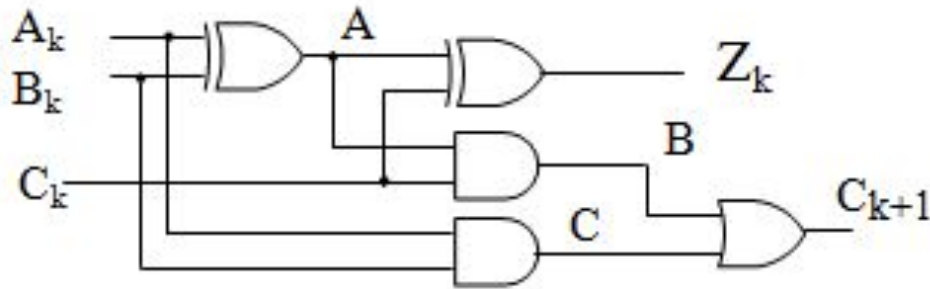
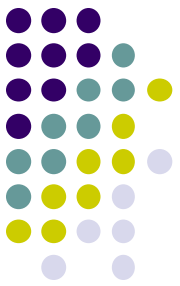
- $S6 = D7 \dots S0 = D1$



Minh họa với $C = 0$

- $\neg C = 1$.
- $S_0 = \text{AND}(\neg C, D_1) = \text{AND}(1, D_1) = D_1$
- $S_1 = \text{OR}(A_0, A_{21})$
 - $A_0 = \text{AND}(C, D_0) = \text{AND}(0, D_0) = 0$
 - $A_{21} = \text{AND}(\neg C, D_2) = \text{AND}(1, D_2) = D_2$
- $S1 = \text{OR}(0, D_2) = D_2$
- $S7 = A62 = \text{AND}(C, D6) = \text{AND}(0, D6) = 0$
- Ứng $C = 0$ là dịch phải. $C = 1$ là dịch trái

Bộ cộng đầy đủ được xây dựng từ các cổng logic cơ bản



Trong đó:

A_k, B_k : đầu vào

Z_k : Kết quả đầu ra

C_k : nhớ từ số thứ $k-1$ sang

C_{k+1} : nhớ tới số thứ $k+1$.

0 0 1 1

0 0 1 1

0 1 0 1

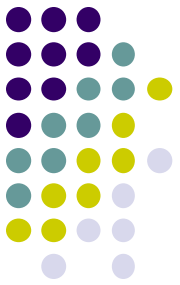
0 1 10 11

Bảng chân lý của bộ cộng đầy đủ:

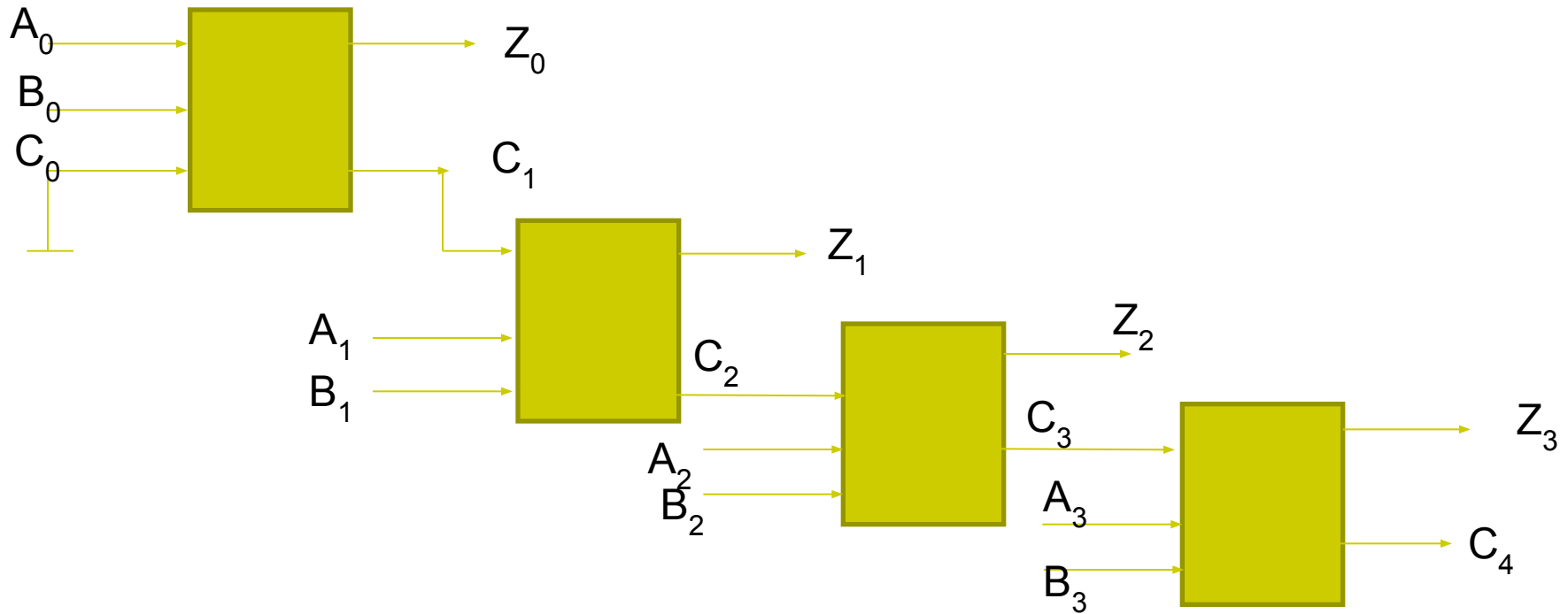
Đầu vào			Đầu ra	
C_k	A_k	B_k	Z_k	C_{k+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

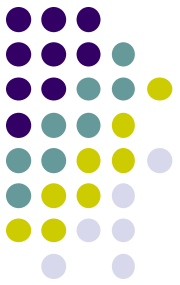
$A = 0101$

$B = 1010$



- $A = 0101 = A_3 A_2 A_1 A_0$
- $B = 1010 = B_3 B_2 B_1 B_0$





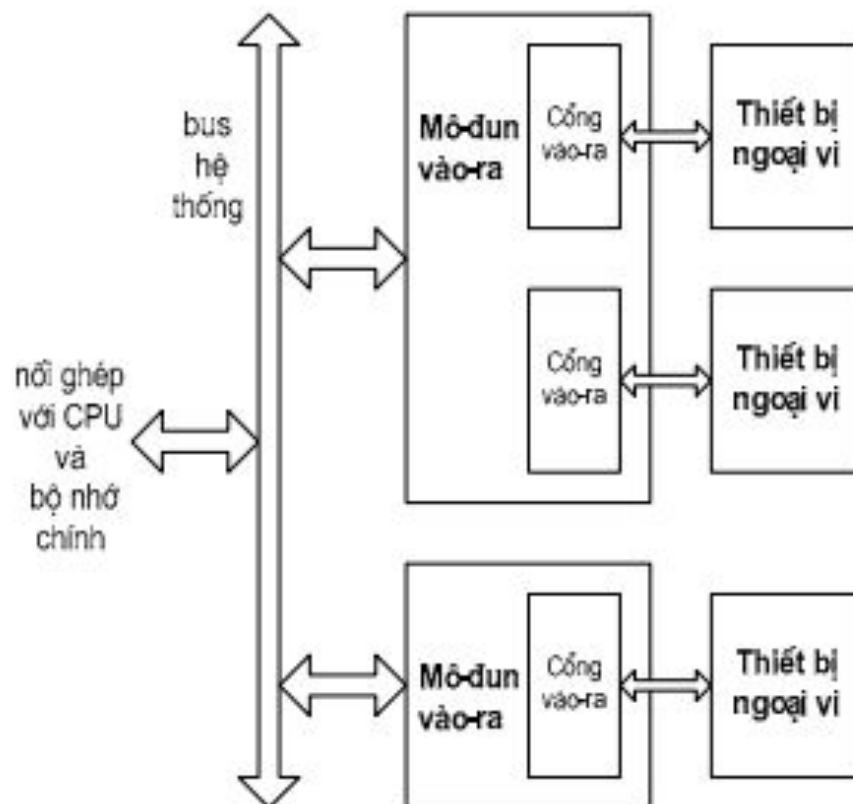
Đường truyền (Bus)

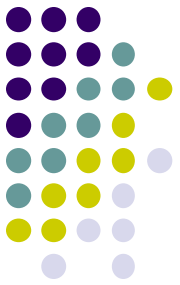
- Có 2 kiểu đường truyền
 - Đường truyền song song
 - Đường truyền nối tiếp
- Có 3 loại bus
 - Bus địa chỉ (Address bus): Độ rộng bus địa chỉ (số đường dây) quyết định dung lượng bộ nhớ mà CPU có thể quản lý được: 2^n .
 - Bus dữ liệu (Data bus): Độ rộng quyết định là trao đổi dữ liệu bao nhiêu bits tại một thời điểm
 - Bus điều khiển (Control bus)

Hệ thống vào/ra



- Cấu trúc cơ bản của hệ thống vào/ra:

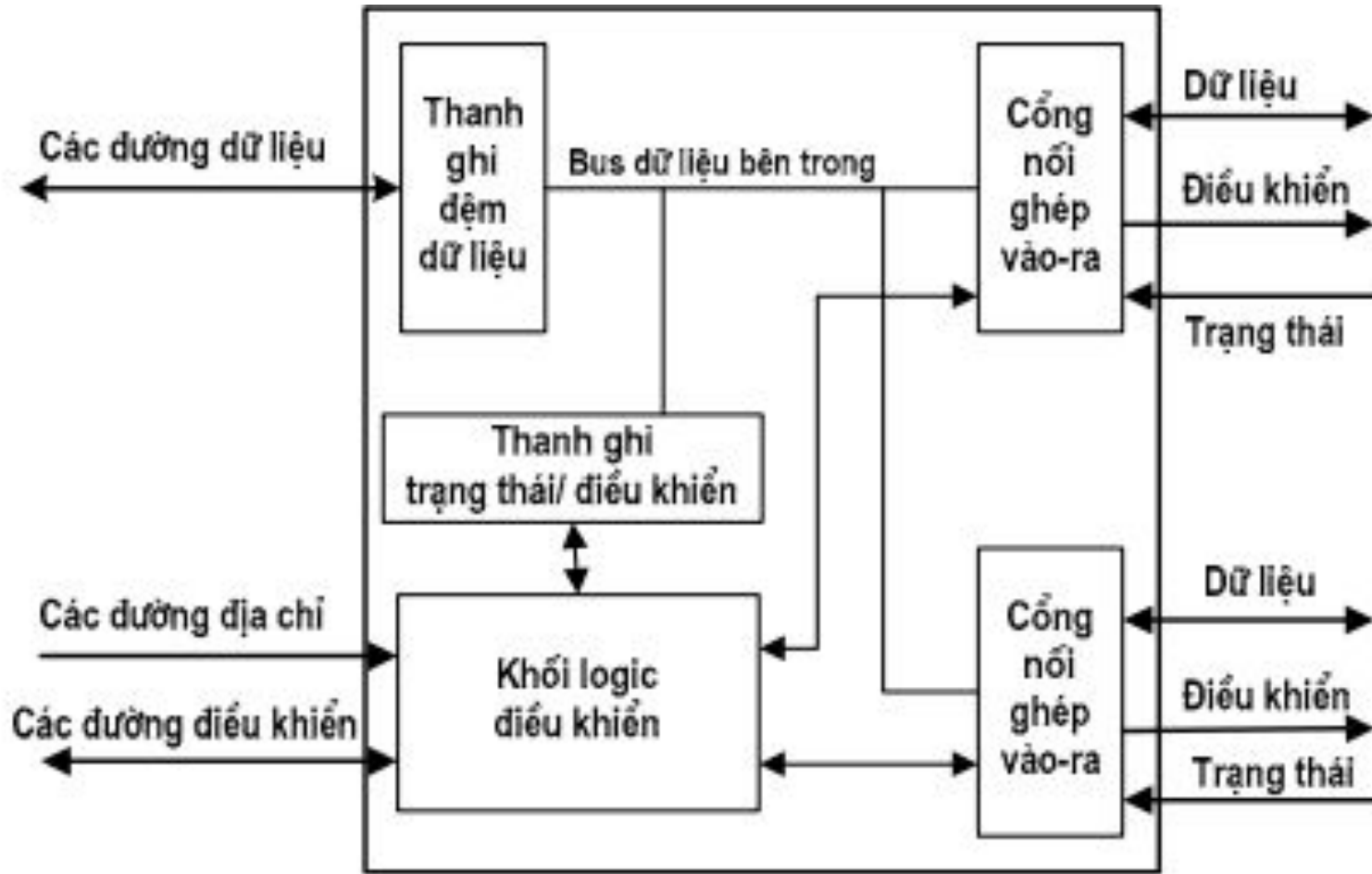
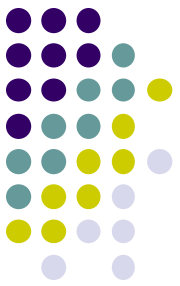




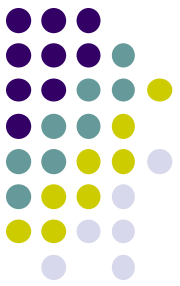
Module vào/ra

- Chức năng của mô-đun vào-ra:
 - Điều khiển và định thời
 - Trao đổi thông tin với CPU
 - Trao đổi thông tin với thiết bị ngoại vi
 - Đệm giữa bên trong máy tính với thiết bị ngoại vi
 - Phát hiện lỗi của thiết bị ngoại vi

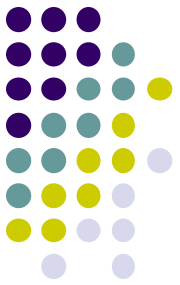
Cấu trúc chung của module vào/ra



Các thành phần của module vào/ra

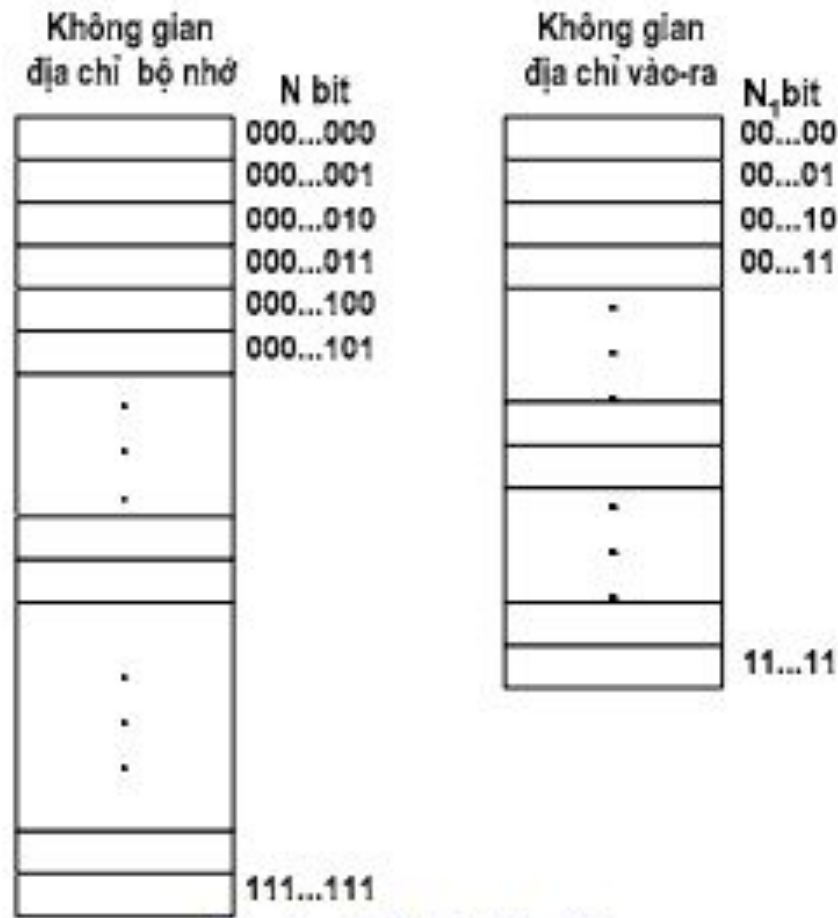


- Thanh ghi đếm dữ liệu: Đếm dữ liệu trong quá trình trao đổi.
- ☐ Các cổng vào-ra (I/O Port): Kết nối với thiết bị ngoại vi, mỗi cổng có một địa chỉ xác định.
- Thanh ghi trạng thái/điều khiển: Lưu giữ thông tin trạng thái/điều khiển cho các cổng vào-ra.
- Khối logic điều khiển: điều khiển mô đun vào/ra.



Địa chỉ hóa cổng vào/ra

- Không gian địa chỉ của CPU



Không gian địa chỉ CPU (tiếp)



- Một số bộ xử lý chỉ quản lý duy nhất một không gian địa chỉ:
 - □ không gian địa chỉ bộ nhớ: 2^N địa chỉ
- □ Ví dụ: Các bộ xử lý 680x0 (Motorola)



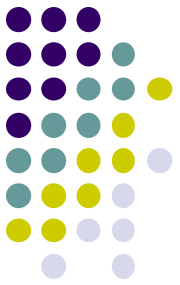
Không gian địa chỉ CPU (tiếp)

- Một số bộ xử lý quản lý hai không gian địa chỉ tách biệt:
 - ☐ Không gian địa chỉ bộ nhớ: 2^N địa chỉ
 - Không gian địa chỉ vào-ra: 2^{N1} địa chỉ
 - Có tín hiệu điều khiển phân biệt truy nhập không gian địa chỉ
 - Tập lệnh có các lệnh vào-ra chuyên dụng
- ☐ Ví dụ: Pentium (Intel)
 - ☐ Không gian địa chỉ bộ nhớ = 2^{32} byte = 4GB
 - ☐ Không gian địa chỉ vào-ra = 2^{16} byte = 64KB
 - Tín hiệu điều khiển /IO/M
 - ☐ Lệnh vào-ra chuyên dụng: IN, OUT

Các phương pháp địa chỉ hóa cổng vào/ra



- Vào-ra riêng biệt
(Isolated IO hay IO mapped IO)
- Vào-ra theo bản đồ bộ nhớ
(Memory mapped IO)



Vào/ra riêng biệt

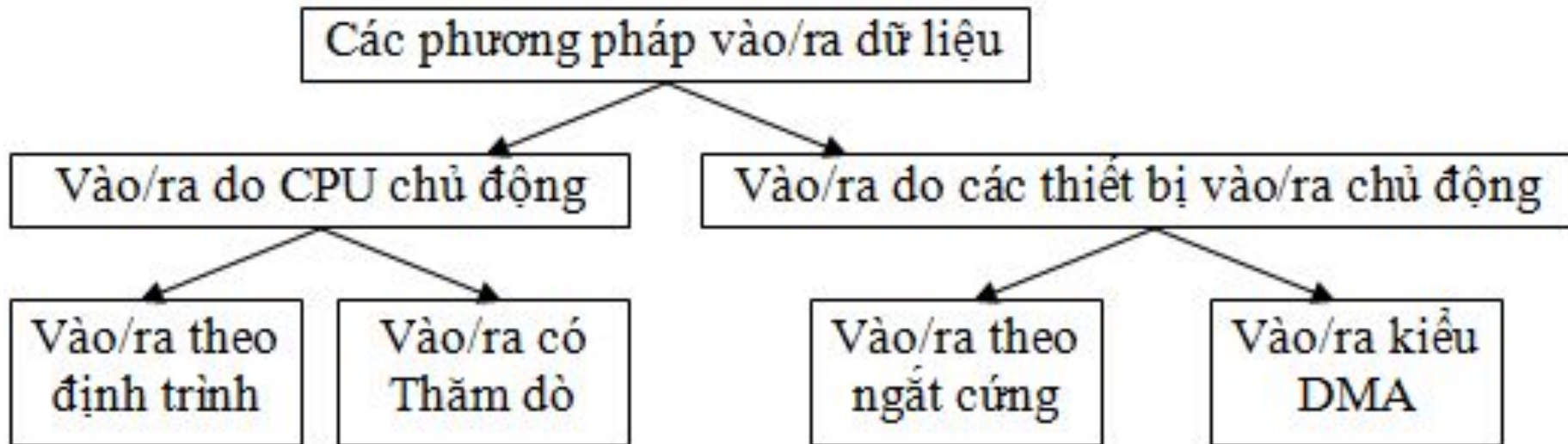
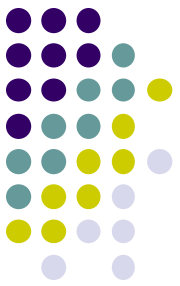
- Cổng vào-ra được đánh địa chỉ theo không gian địa chỉ vào-ra
- CPU trao đổi dữ liệu với cổng vào-ra thông qua các lệnh vào-ra chuyên dụng (IN, OUT)
- Chỉ có thể thực hiện trên các hệ thống có quản lý không gian địa chỉ vào-ra riêng biệt.

Vào/ra theo bản đồ bộ nhớ

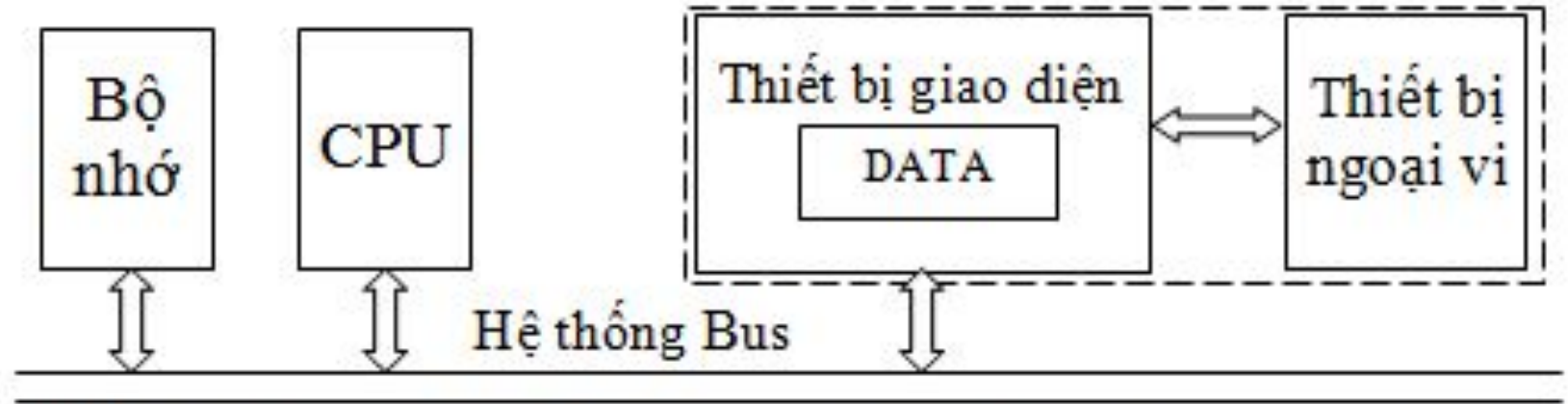


- Cổng vào-ra được đánh địa chỉ theo không gian địa chỉ bộ nhớ
- Vào-ra giống như đọc/ghi bộ nhớ
- CPU trao đổi dữ liệu với cổng vào-ra thông qua các lệnh truy nhập dữ liệu bộ nhớ
- Có thể thực hiện trên mọi hệ thống

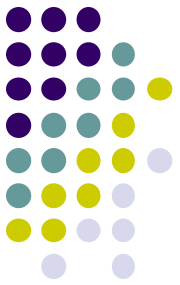
Các phương pháp điều khiển vào/ra



Vào/ra theo định trình

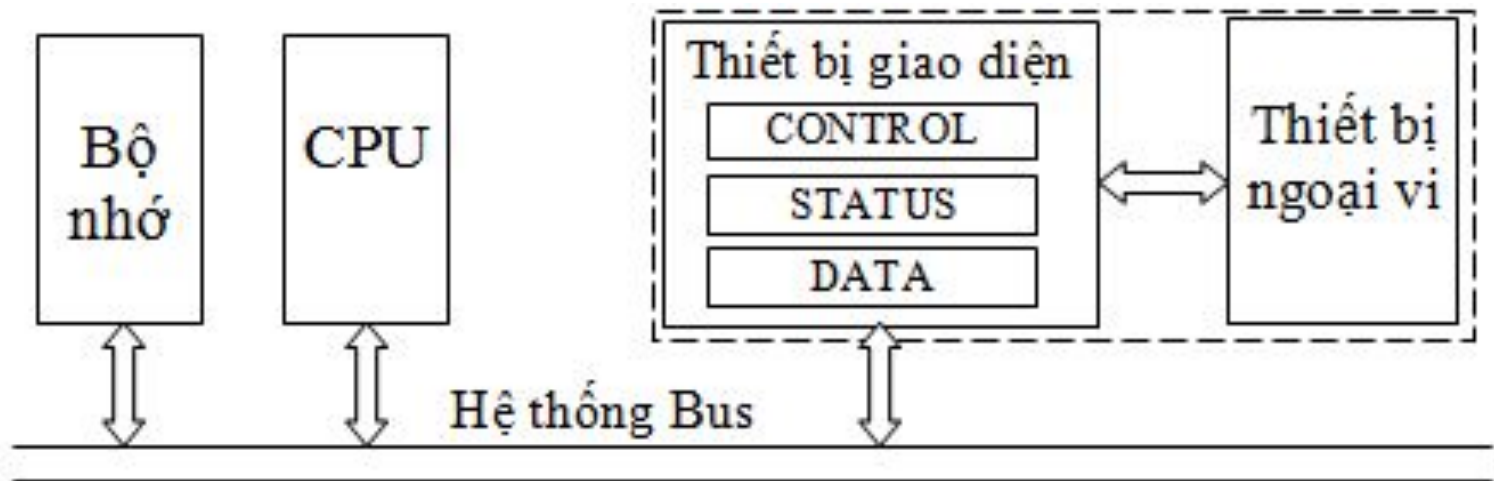


- Thực hiện tức thời vào ra bằng cách sử dụng câu lệnh vào/ra (IN/OUT).
- CPU không cần kiểm tra trạng thái sẵn sàng của thiết bị cũng như mô đun vào/ra.

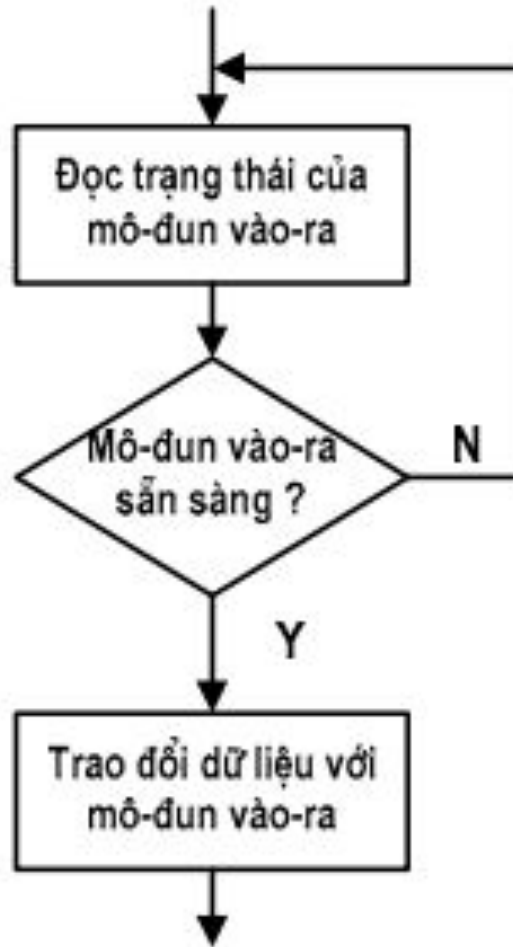
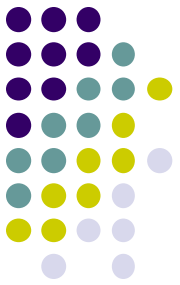


Vào/ra có thăm dò

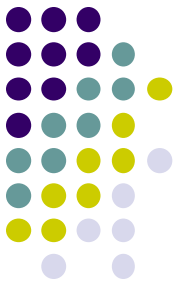
- Sơ đồ ghép nối



Lưu đồ đoạn chương trình vào/ra có thăm dò

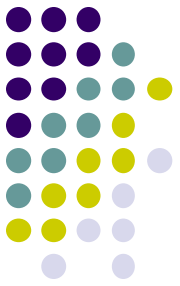


Hoạt động vào/ra có thăm dò



- CPU yêu cầu thao tác vào-ra
- Mô-đun vào-ra thực hiện thao tác
- Mô-đun vào-ra thiết lập các bit trạng thái
- CPU kiểm tra các bit trạng thái:
 - Nếu chưa sẵn sàng thì quay lại kiểm tra
 - Nếu sẵn sàng thì chuyển sang trao đổi dữ liệu với mô-đun vào-ra.

Đặc điểm vào/ra có thăm dò



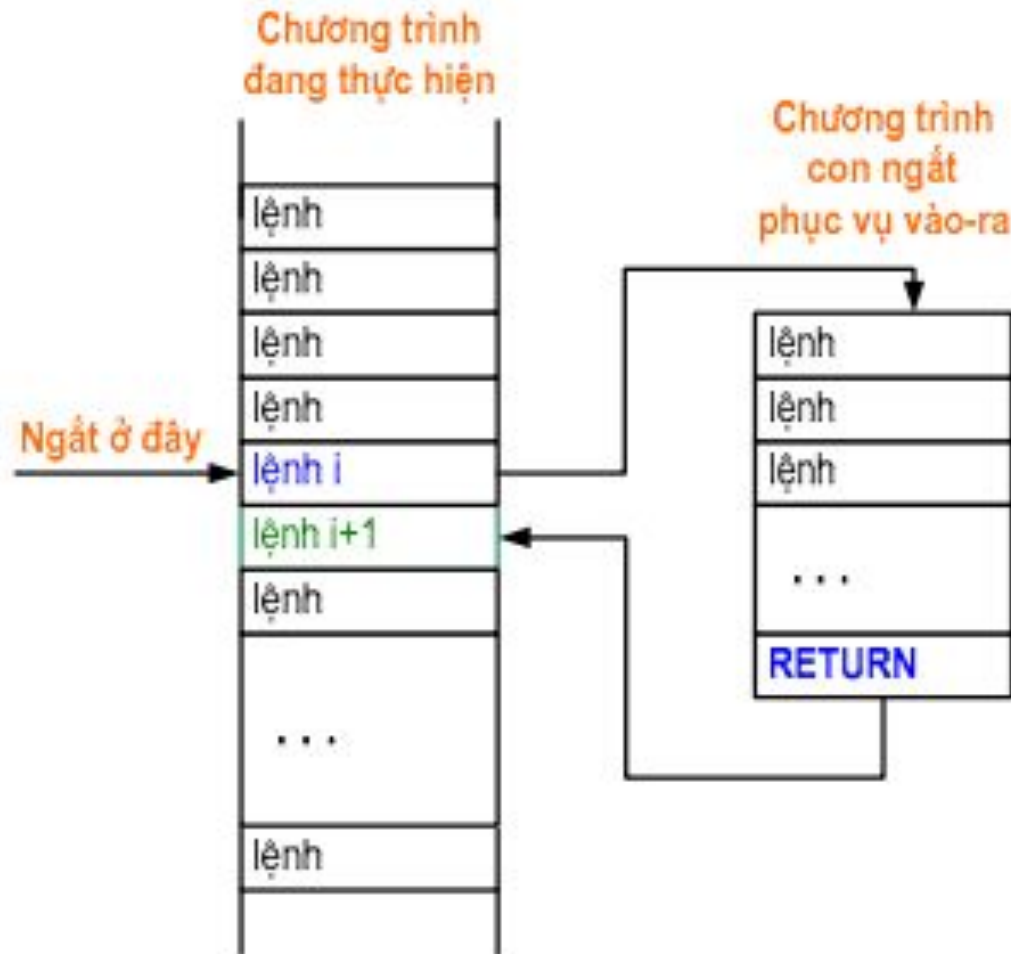
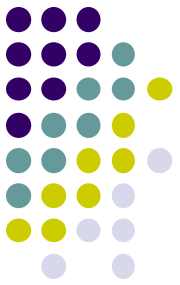
- CPU trực tiếp điều khiển vào-ra
- □ CPU đợi mô-đun vào-ra → tiêu tốn thời gian của CPU.

Vào/ra bằng ngắt cứng



- Nguyên tắc chung:
 - CPU không phải đợi trạng thái sẵn sàng của mô-đun vào-ra, CPU thực hiện một chương trình nào đó.
 - Khi mô-đun vào-ra sẵn sàng thì nó phát tín hiệu ngắt CPU
 - CPU thực hiện chương trình con vào-ra tương ứng để trao đổi dữ liệu
 - CPU trở lại tiếp tục thực hiện chương trình đang bị ngắt.

Chuyển điều khiển đến chương trình con phục vụ ngắt

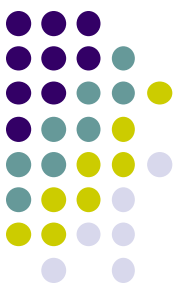


Hoạt động vào dữ liệu: Nhìn từ mô đun vào/ra



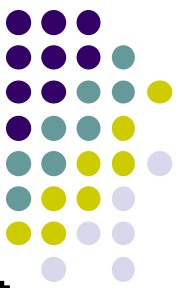
- Mô-đun vào-ra nhận tín hiệu điều khiển *đọc* từ CPU.
- Mô-đun vào-ra nhận dữ liệu từ thiết bị ngoại vi, trong khi đó CPU làm việc khác.
- ☐ Khi đã có dữ liệu → mô-đun vào-ra phát tín hiệu ngắt CPU.
- CPU yêu cầu dữ liệu.
- Mô-đun vào-ra chuyển dữ liệu đến CPU.

Hoạt động vào dữ liệu: Nhìn từ CPU



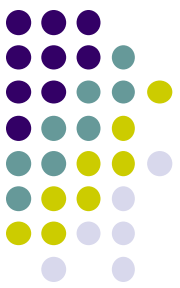
- Phát tín hiệu điều khiển **đọc**
- Làm việc khác
- Cuối mỗi chu trình lệnh, kiểm tra tín hiệu ngắt
- Nếu bị ngắt:
 - Cắt ngữ cảnh (nội dung các thanh ghi)
 - Thực hiện chương trình con ngắt để vào dữ liệu
 - Khôi phục ngữ cảnh của chương trình đang thực hiện

Các vấn đề nảy sinh khi thiết kế



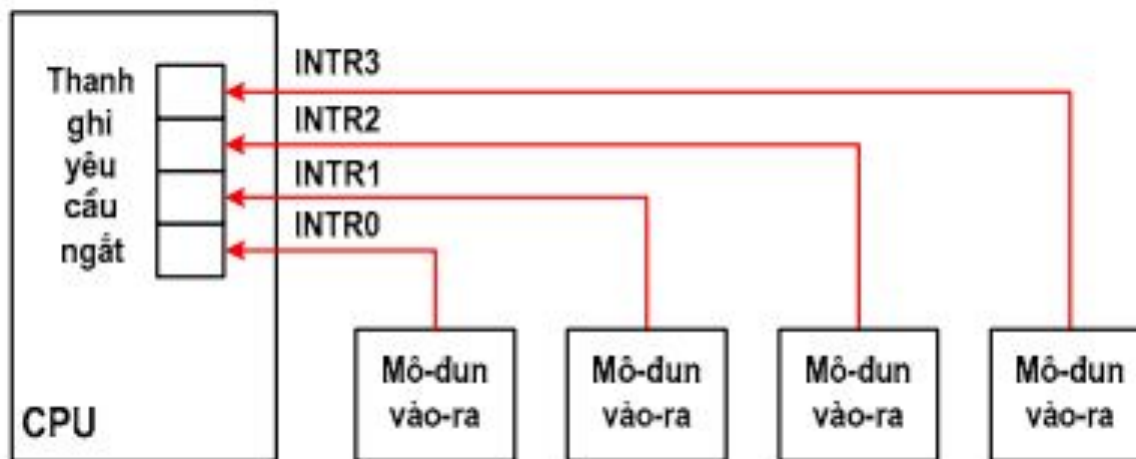
- Làm thế nào để xác định được mô-đun vào-ra nào phát tín hiệu ngắt ?
- CPU làm như thế nào khi có nhiều yêu cầu ngắt cùng xảy ra ?

Các phương pháp ghép nối ngắt



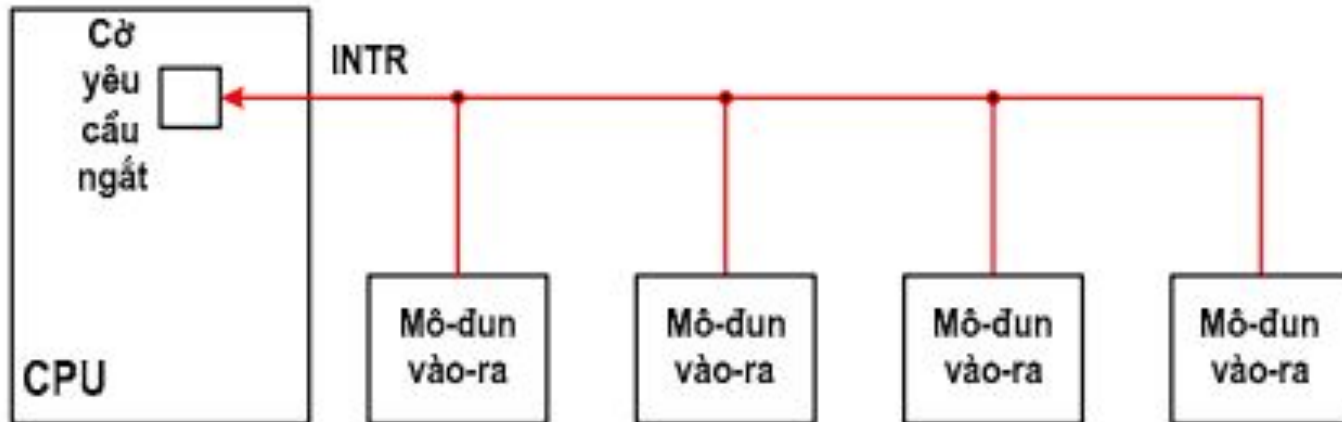
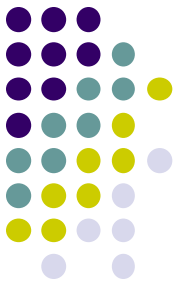
- ☐ Sử dụng nhiều đường yêu cầu ngắt
- Hỏi vòng bằng phần mềm (Software Poll)
- Hỏi vòng bằng phần cứng (Daisy Chain or Hardware Poll)
- Sử dụng bộ điều khiển ngắt (PIC)

Sử dụng nhiều đường dây yêu cầu ngắt



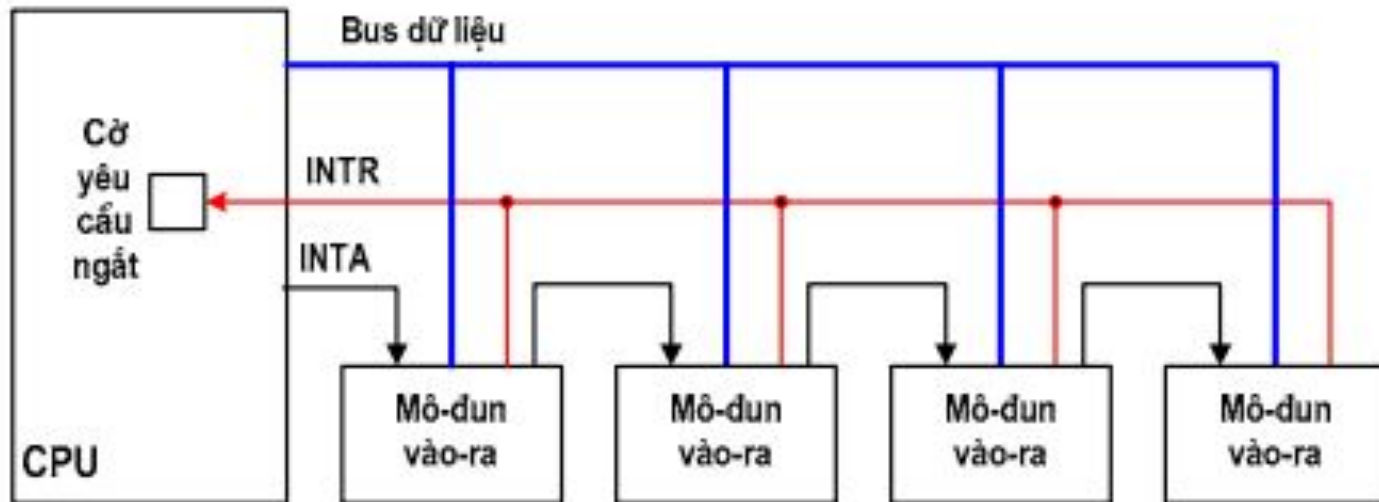
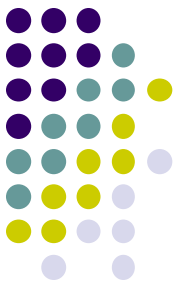
- Mỗi mô-đun vào-ra được nối với một đường yêu cầu ngắt
- CPU phải có nhiều đường tín hiệu yêu cầu ngắt
- Hạn chế số lượng mô-đun vào-ra
- Các đường ngắt được quy định mức ưu tiên

Hỏi vòng bằng phần mềm



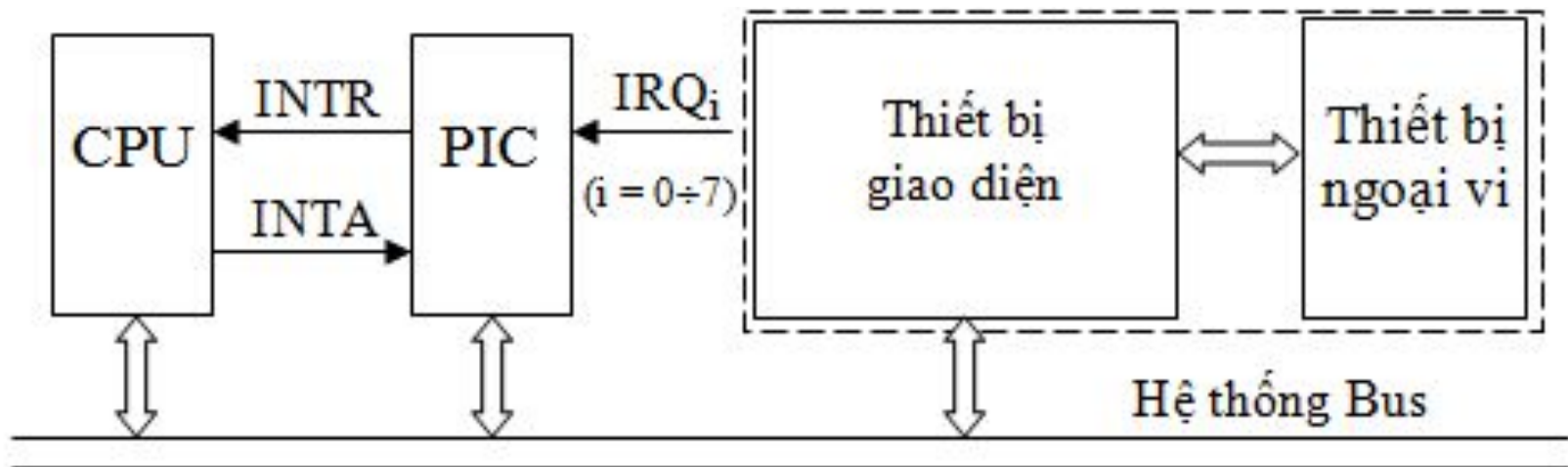
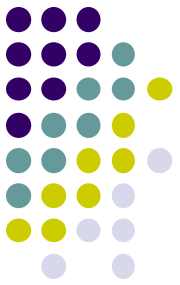
- CPU thực hiện phần mềm hỏi lần lượt từng mô-đun vào-ra
- Chậm
- Thứ tự các mô-đun được hỏi vòng chính là thứ tự ưu tiên

Hỏi vòng bằng phần cứng

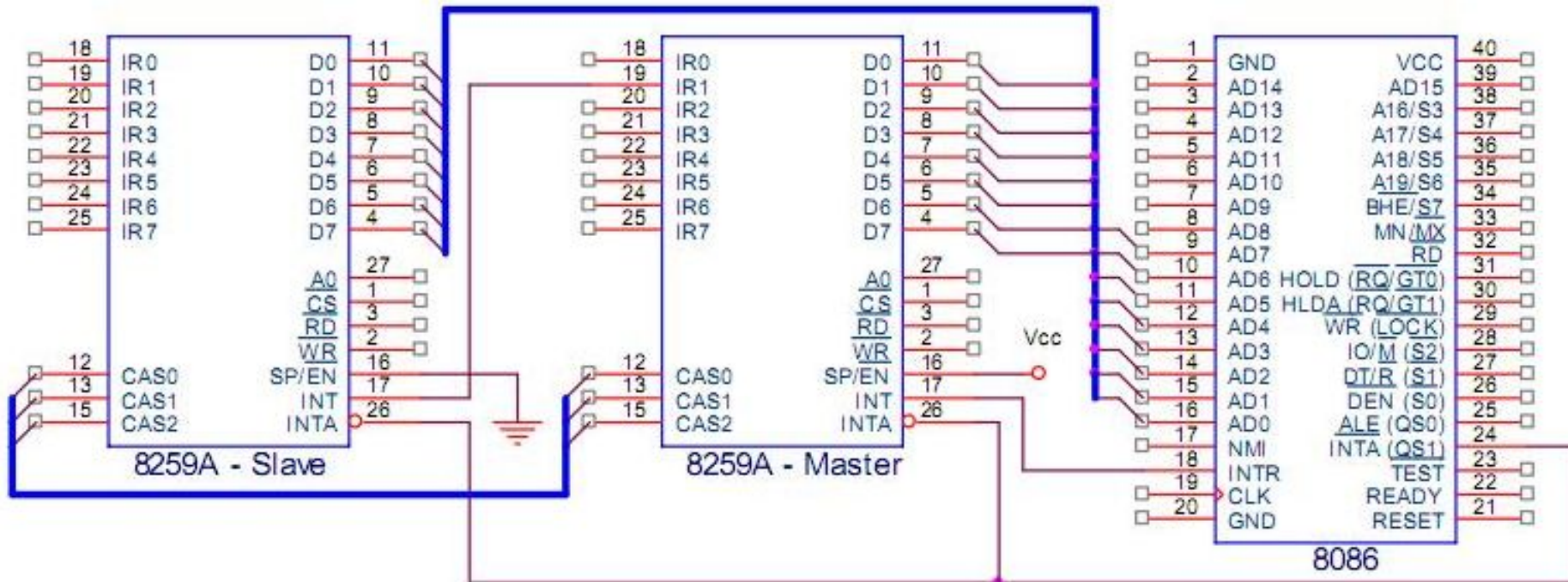
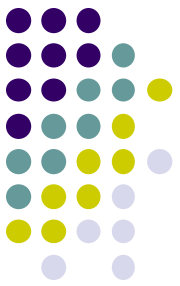


- CPU phát tín hiệu chấp nhận ngắt (INTA) đến mô-đun vào-ra đầu tiên.
- Nếu mô-đun vào-ra đó không gây ra ngắt thì nó gửi tín hiệu đến mô-đun kế tiếp cho đến khi xác định được mô-đun gây ngắt.
- Thứ tự các mô-đun vào-ra kết nối trong chuỗi xác định thứ tự ưu tiên.

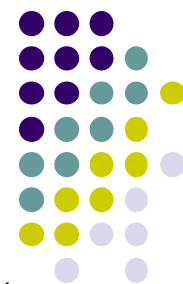
Sử dụng vi mạch điều khiển ngắt lập trình được (PIC)



Ghép tầng 2 vi mạch PIC



Hoạt động của vào ra bằng ngắt cứng



- Các thiết bị vào/ra có yêu cầu phục vụ, phát ra tín hiệu IRQ_i ($i = 0 \div 7$) tới PIC. Mỗi thiết bị vào/ra đã được ấn định sẵn một **số hiệu ngắt** nhất định.
- Thiết bị PIC lựa chọn **số hiệu ngắt** có mức ưu tiên cao nhất, phát tín hiệu INTR (Interrupt Request) tới CPU yêu cầu CPU phục vụ.
- Nếu CPU chấp nhận ngắt, CPU hoàn thành nốt lệnh đang thực hiện, tiến hành lưu trạng thái của tiến trình đang thực hiện và trạng thái hiện thời của CPU.

Hoạt động của vào ra bằng ngắt cứng (tiếp)



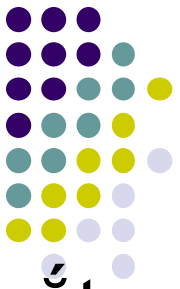
- CPU phát ra tín hiệu INTA (Interrupt Acknowledge) trả lời PIC, báo sẵn sàng phục vụ yêu cầu ngắt.
- PIC phát ra **số hiệu ngắt** được chọn tới CPU qua Bus dữ liệu.
- Dựa vào số hiệu ngắt này, CPU xác định được địa chỉ của chương trình con phục vụ ngắt, kích hoạt và thực hiện chương trình con phục vụ ngắt để thực hiện vào/ra dữ liệu với thiết bị được chọn
- Khi chương trình con phục vụ ngắt kết thúc, CPU lấy lại trạng thái cũ của CPU và của tiến trình vừa bị ngắt để tiếp tục thi hành.

Đặc điểm của vào/ra điều khiển bằng ngắt cứng



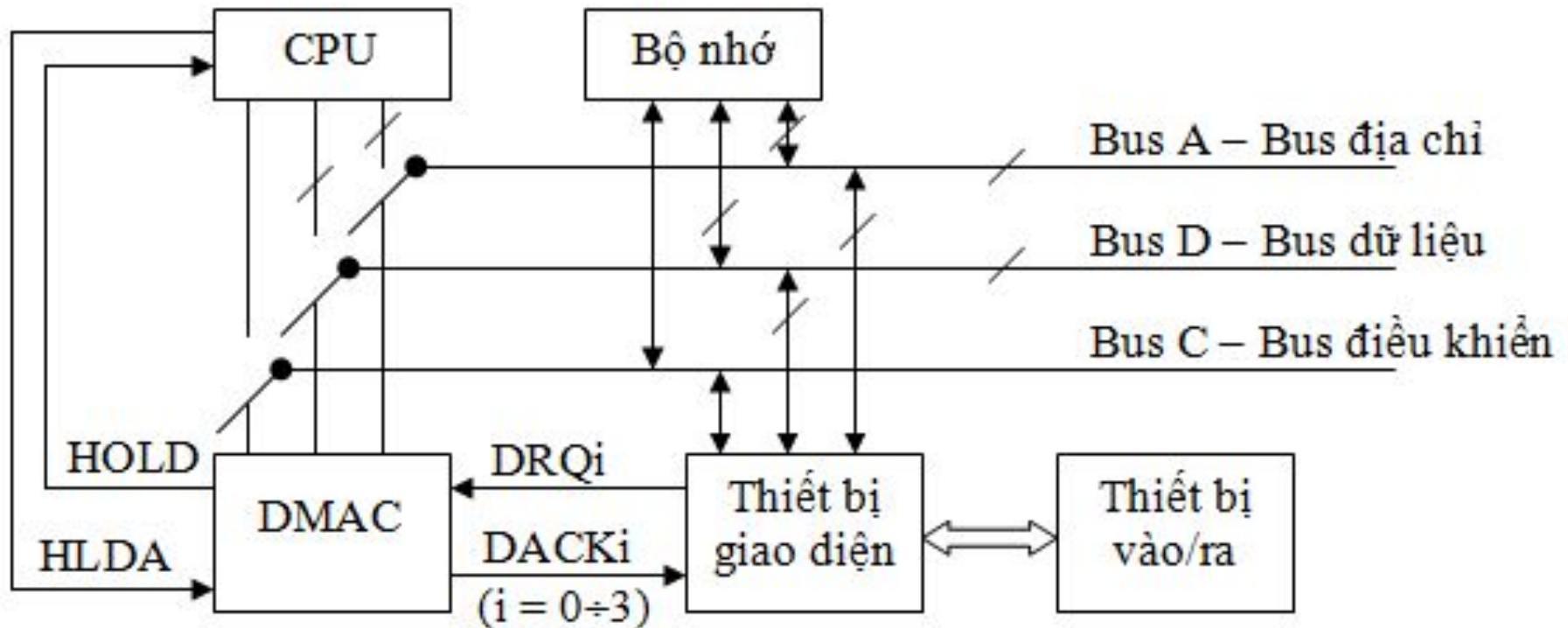
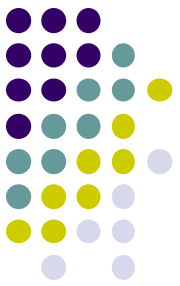
- Có sự kết hợp giữa phần cứng và phần mềm
 - Phần cứng: gây ngắt CPU
 - Phần mềm: trao đổi dữ liệu
- CPU trực tiếp điều khiển vào-ra
- ☐ CPU không phải đợi mô-đun vào-ra → hiệu quả sử dụng CPU tốt hơn

Vào/ra truy nhập bộ nhớ trực tiếp (DMA: Direct Memory Access)

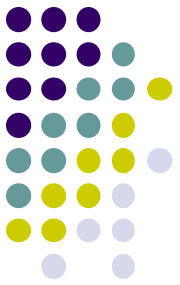


- Vào-ra bằng chương trình và bằng ngắt do CPU trực tiếp điều khiển:
 - Chiếm thời gian của CPU
 - Tốc độ truyền bị hạn chế vì phải chuyển qua CPU
- Để khắc phục dùng DMA
 - Thêm mô-đun phần cứng trên bus → DMAC (Controller)
 - DMAC điều khiển trao đổi dữ liệu giữa mô-đun vào-ra với bộ nhớ chính

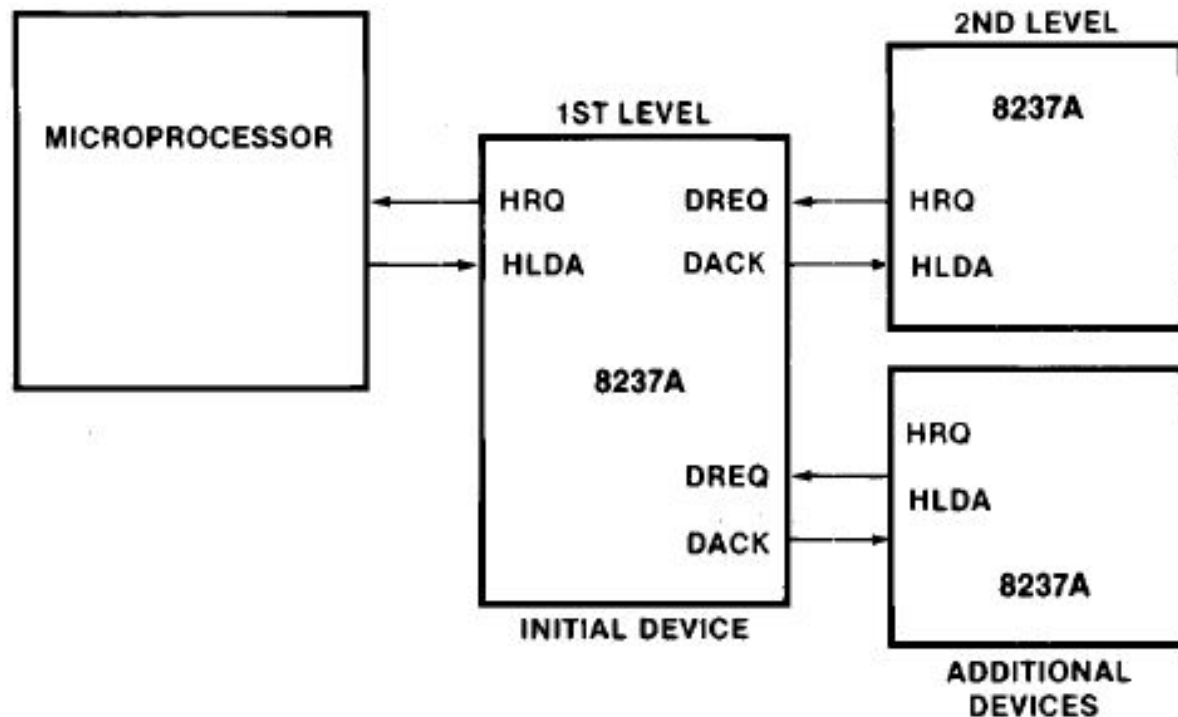
Ghép nối DMAC và bus hệ thống



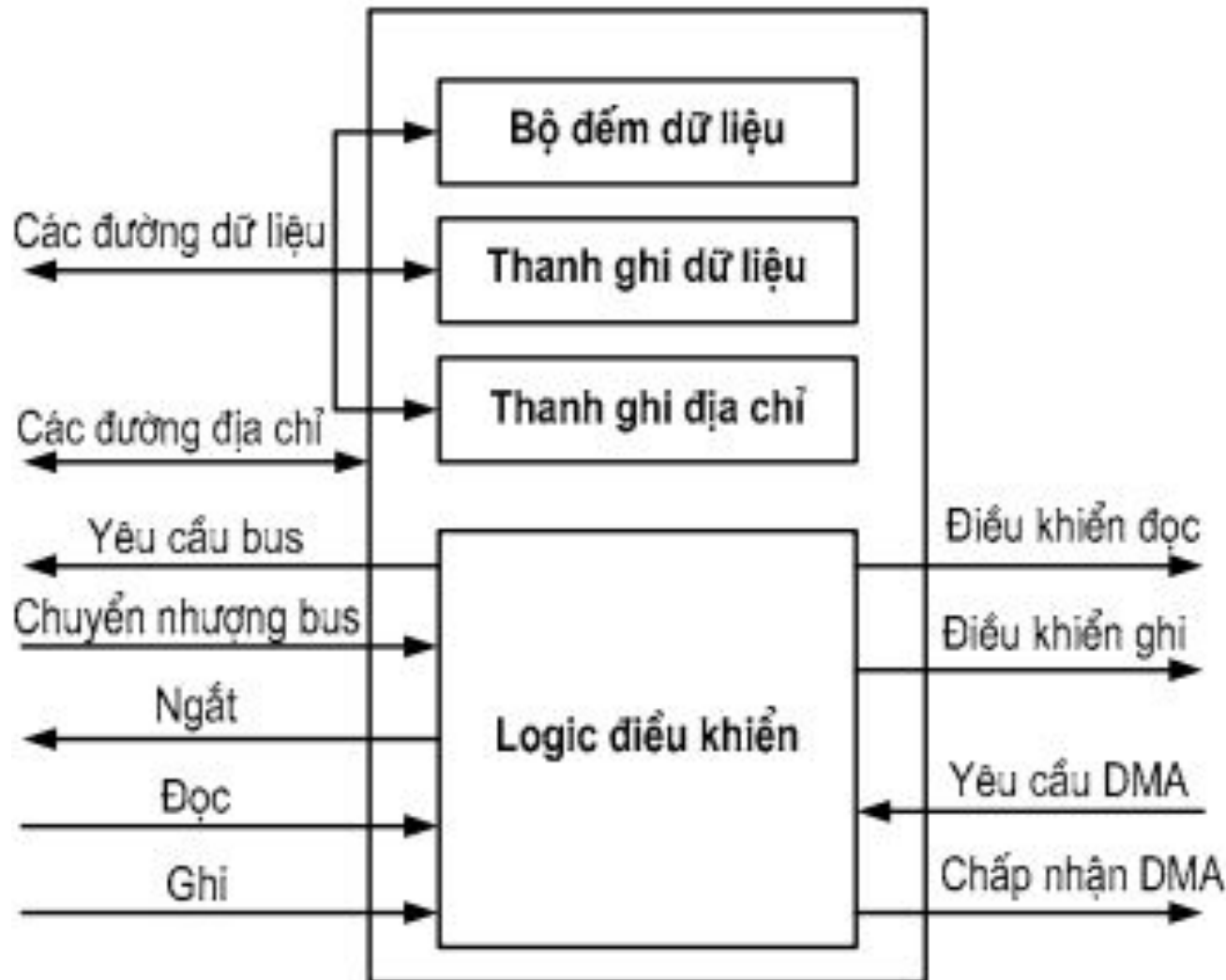
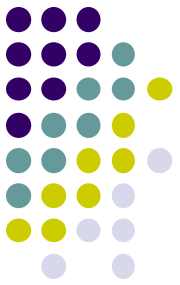
Nối tầng DMAC



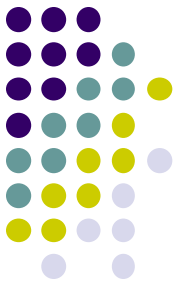
- Mỗi DMAC phục vụ tối đa cho 4 thiết bị vào/ra theo kiểu DMA → nhiều cần nối tầng.



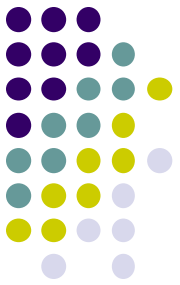
Sơ đồ cấu trúc của DMAC



Các thành phần của DMAC

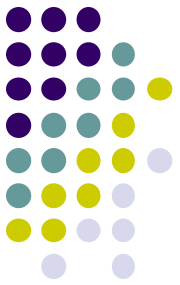


- Thanh ghi dữ liệu: chứa dữ liệu trao đổi
- Thanh ghi địa chỉ: chứa địa chỉ ngăn nhớ dữ liệu
- Bộ đếm dữ liệu: chứa số từ dữ liệu cần trao đổi
- Logic điều khiển: điều khiển hoạt động của DMAC



Hoạt động DMA

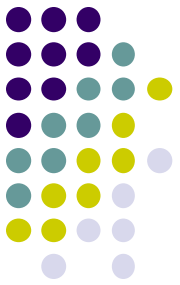
- Thiết bị có nhu cầu trao đổi DMA sẽ gửi yêu cầu đến chân DRQ_i cho DMAC, DMAC chọn thiết bị có mức ưu tiên cao nhất.
- DMAC phát tín hiệu BRQ (Bus Request)/HOLD = 1 cho CPU, yêu cầu CPU chuyển nhượng Bus.
- Nếu CPU đồng ý thì thực hiện nốt chu kỳ bus, CPU sẽ tiến hành khởi tạo cho DMAC



Hoạt động DMA (tiếp)

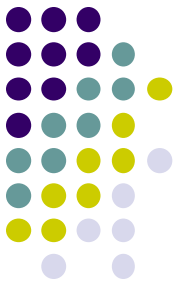
- DMAC điều khiển bus. Phát tín hiệu DACK_i để báo cho thiết bị biết là chấp nhận trao đổi DMA.
- DMAC điều khiển quá trình trao đổi dữ liệu giữa thiết bị vào/ra và bộ nhớ.
- Sau khi truyền được một từ dữ liệu thì:
 - Nội dung thanh ghi địa chỉ tăng
 - Nội dung bộ đếm dữ liệu giảm
- Khi bộ đếm dữ liệu = 0, DMAC gửi tín hiệu ngắt CPU để báo kết thúc DMA.

Các kiểu thực hiện DMA

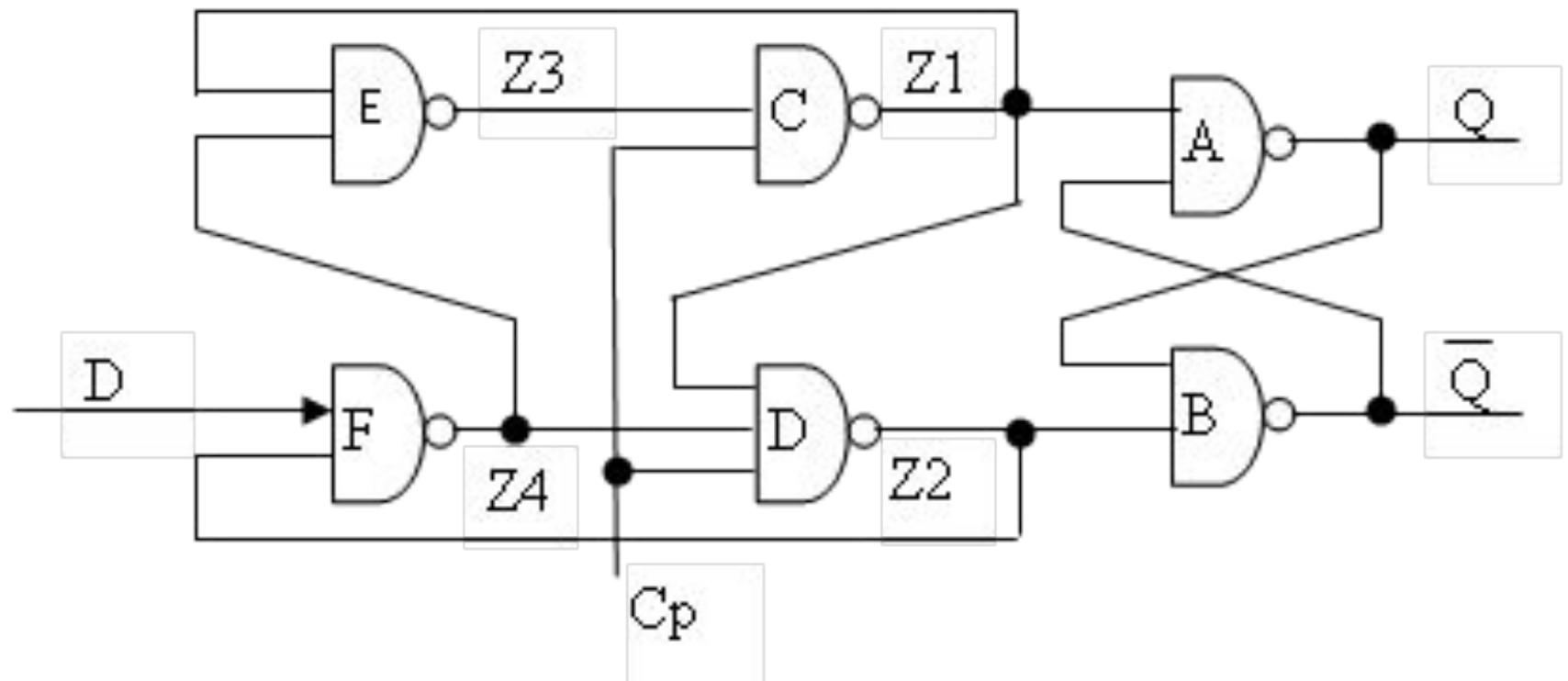
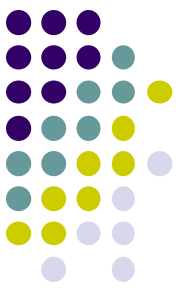


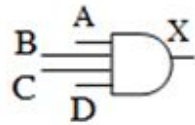
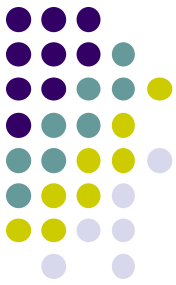
- DMA truyền theo khối (Block-transfer DMA): DMAC sử dụng bus để truyền xong cả khối dữ liệu.
- DMA lấy chu kỳ (Cycle Stealing DMA): DMAC cưỡng bức CPU treo tạm thời từng chu kỳ bus, DMAC chiếm bus thực hiện truyền một từ dữ liệu.
- DMA trong suốt (Transparent DMA): DMAC nhận biết những chu kỳ nào CPU không sử dụng bus thì chiếm bus để trao đổi một từ dữ liệu.

Đặc điểm DMA

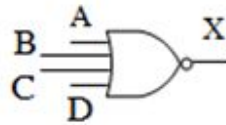


- CPU không tham gia trong quá trình trao đổi dữ liệu.
- DMAC điều khiển trao đổi dữ liệu giữa bộ nhớ chính với mô-đun vào-ra (hoàn toàn bằng phần cứng) → tốc độ nhanh.
- Phù hợp với các yêu cầu trao đổi mảng dữ liệu có kích thước lớn (các ổ đĩa).

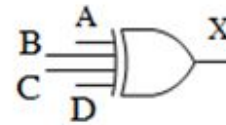




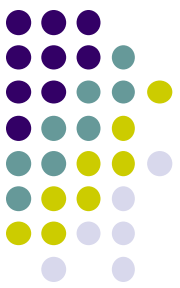
in				out
A	B	C	D	X
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	



in				out
A	B	C	D	X
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

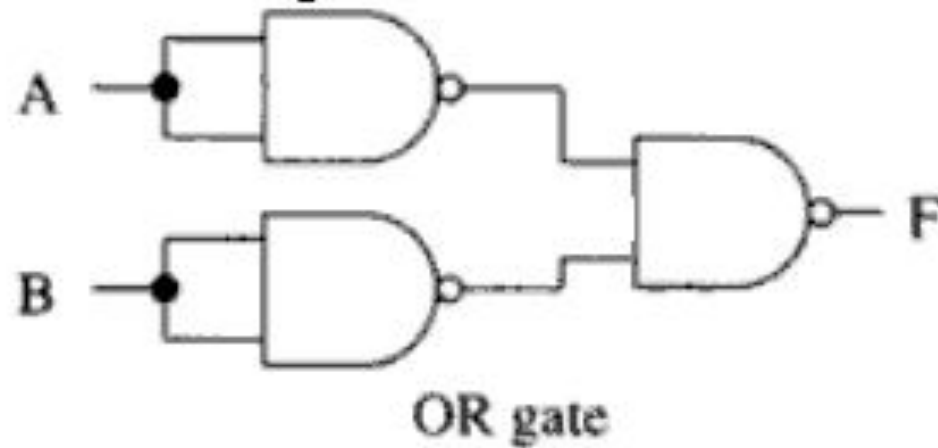


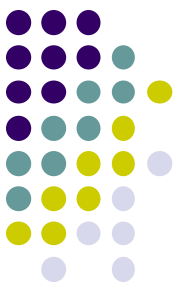
in				out
A	B	C	D	X
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	



Câu 3.26.

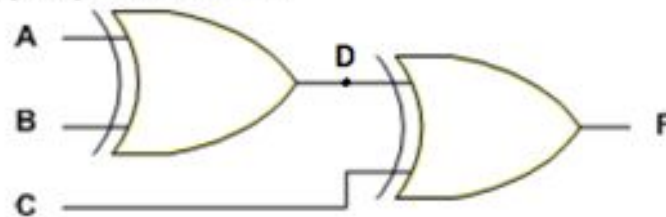
Chứng minh rằng mạch sau là cổng OR 2 đầu vào

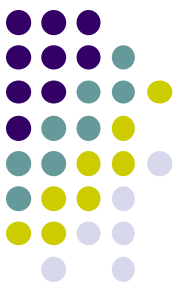




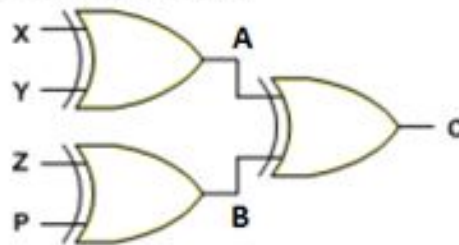
Câu 3.30.

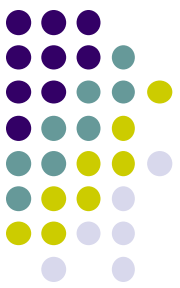
Chứng minh rằng mạch sau là mạch lẻ 3 đầu vào (số bit 1 là lẻ thì đầu ra là 1, ngược lại đầu ra là 0), với đầu vào là A, B, C ; đầu ra là F





Chứng minh rằng mạch sau là mạch lẻ 4 đầu vào (số bit 1 là lẻ thì đầu ra là 1, ngược lại đầu ra là 0), với đầu vào là X, Y, Z, P ; đầu ra là C





9. Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit:

-0.03125