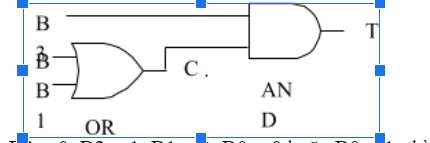
**Câu 2: *(05 điểm):*** Mạch logic được thiết kế để phát hiện lỗi trong mã BCD. Lối vào là 3 bit cao của mã BCD, lối ra ở trạng thái 1 khi có lỗi.

Chứng minh rằng B3 = 0, B2 = 1, B1 = 0, B0 = 0 hoặc B0 = 1, thì T = 0 (mã BCD không lỗi)

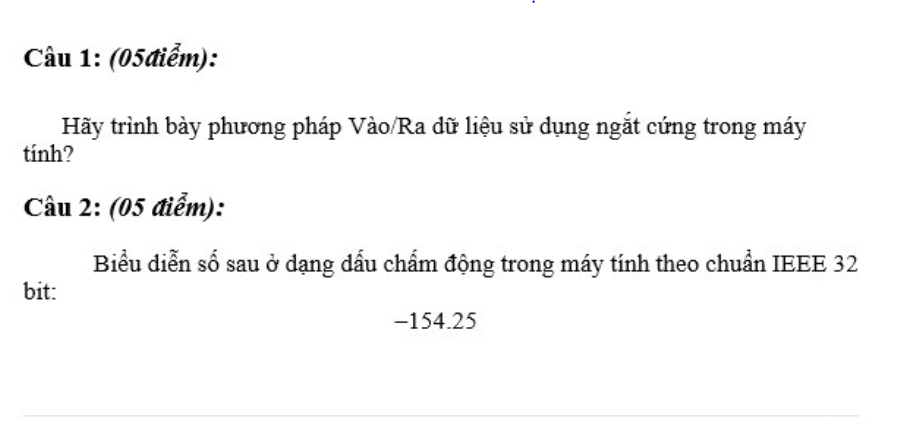
C = OR(B1,B2)=1

T = AND(C,B3)=(1,0)=0

* Mã BCD không lỗi

BCD là mà Gồm 4 bits, biểu diễn từ 0-9, các từ 10-15 là mã BCD lỗi

VD: mã



Câu 1:

- KN ngắt cứng: là CPU phải tạm dừng tiến trình đang thực hiện để chuyển sang thực hiện tiến trình phục vụ ngắt khi có yêu cầu ngắt từ phần mạch bên ngoài gửi đến CPU.

- Phương pháp vào/ra theo ngắt cứng là phương pháp mà thiết bị vào/ra chủ động khởi tạo quá trình vào/ra dữ liệu nhờ hệ thông ngắt.

-Ưu điểm : Quá trình vào/ra dữ liệu được thực hiện ngay khi có yêu cầu, độ tin cậy cao. Hiệu suất CPU cao vì không mất thời gian thăm dò trạng thái làm việc của thiết bị vào/ra.

-Nhược điểm : phương pháp này vẫn phải thông qua CPU nên hiệu năng hoạt động toàn hệ thống chưa phải là cao nhất.

Câu2:

Bước 1 : Công thức: R =(-1)^S \* 1.M \* 2^(E-Bias)

* S là bit dấu, chiếm 1 bit dấu.
* **E** là Exponent (phần mũ) luôn lớn hơn hoặc bằng 0  chiếm 8 bit.
* **M** là Mantissa (phần định trị) - chỉ lưu phần lẻ sau dấu chấm nhị phân, chiếm 23 bit.
* Vì phần mũ (E – Bias) có thể âm, hoặc dương, xong trong cấu trúc dữ liệu lưu trữ số dấu chấm động, chỉ được phép có một bit dấu của số (không được phép thêm bit dấu của số mũ) nên E phải trong khoảng [0 ÷ 255]. Vì vậy để khử giá trị âm cho phần mũ, ta sử dụng số Bias, E < Bias thì phần mũ âm,   
  E > Bias thì phần mũ dương. Giá trị của Bias được xác định để cân đối cả 2 phần âm và dương. Vậy với E chiếm 8 bit thì Bias = 28/2 -1 = 127
* Độ chính xác dữ liệu là 2-127: vì E ≥ 0, nên E – Bias ≥ –127, do vậy giá trị dữ liệu biểu diễn nhỏ nhất với kiểu short real là 2–127.
* Số bít E là 8 với số 32 bit, số bit của E là 11 nếu số thực 64 bits
* **bias =  2^(số bits của E )/2 -1**
* Bias = 2^( Số bit biểu biểu diễn )/2 - 1 = E/2 -1 = 2^8/2 -1 =127

Bước 2: Vẽ khuôn dạng: S E M

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S | E7 |  | E0 | M22 |  | M0 |

Bước 3: Chuyển về bits nhị phân

154 = 10011010

0.25 = 0.01

=>Ghép lại 154.25 = 10011010.01

Bước 3: Chuẩn hóa

R = (-1)^1 \* 1.001101001\* 2^7

Bước 4: Xác định S, E, M

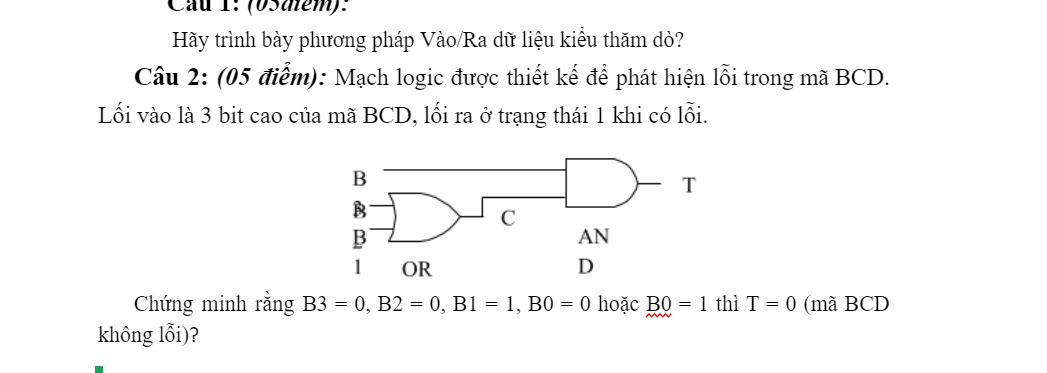
S = 1

E = 127 + 7=134 = 10000110b

M = 00110100100000000000000b

Bước 5: Lắp khuôn:

X = 1100 0011 0001 1010 0100 0000 0000 0000b = C31A4000H



Câu 1 :

Phương pháp vào/ra kiểu 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 sãn sàng thì quay lại kiểm tra

                 + Nếu sãn sàng thì chuyển sang trao đổi dữ liệu với mô-đun vào/ra

* Ưu điểm : quá trình trao đổi dữ liệu có độ tin cậy rất cao vì việc truyền nhận dữ liệu chỉ xảy ra khi hai bên truyền và nhận đều sẵn sàng.
* Nhược điểm : chiếm dụng nhiều thời gian CPU cho việc thăm dò nên hiệu quả hoạt động của hệ thống không cao

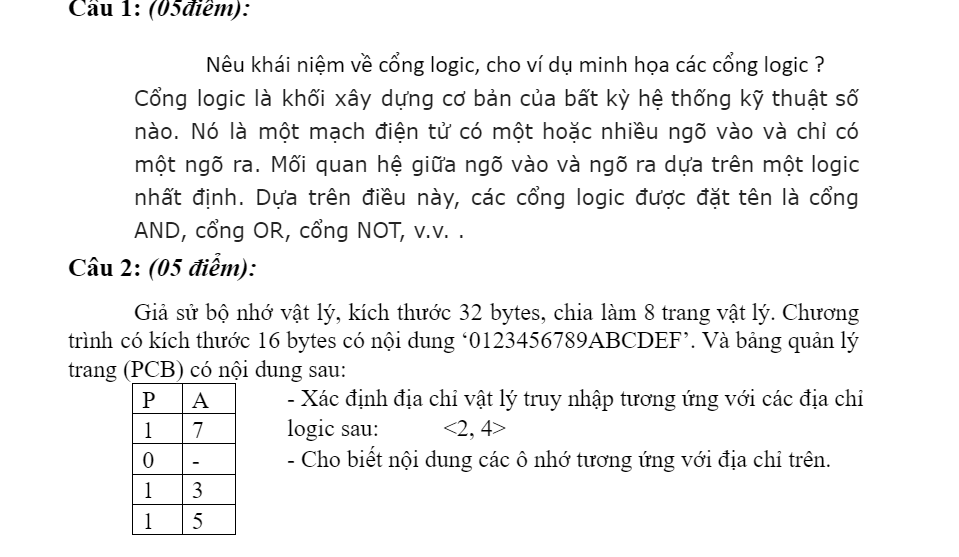
Câu 2:

C = OR(B1,B2)=1

T = AND(C,B3)=(1,0)=0

* Mã BCD không lỗi

BCD là mà dồm 4 bits, biểu diễn từ 0-9, các từ 10-15 là mã BCD lỗi



Câu 2:

* Địa chỉ của lệnh hoặc dữ liệu được biểu diễn là bộ đôi giá trị <Np, d>, địa chỉ này được gọi là địa chỉ logic.
  + Np là số hiệu của trang logic cần truy nhập.
  + d là địa chỉ tương đối của lệnh hay dữ liệu trong trang.

***Bài làm***

***\* Phần tính toán chung***

- Kích thước một trang vật lý là: L = dung lượng bộ nhớ vật lý / số trang vật lý.

= 32/8 = 4 bytes = 22.

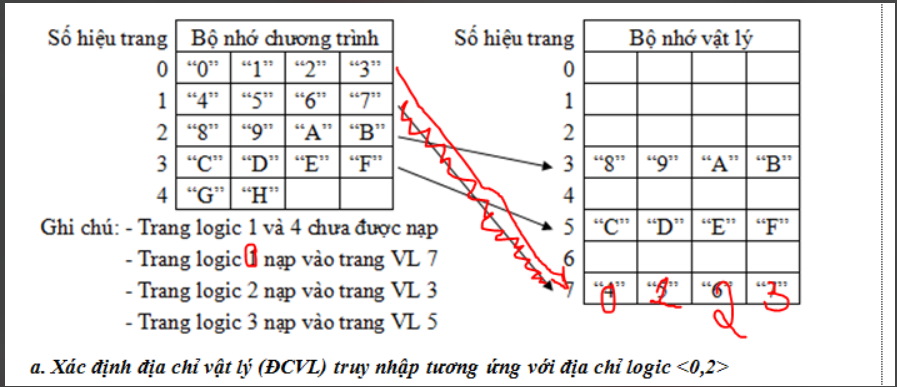
- Vậy số bit để đánh độ lệch trong trang là: n1 = 2.(bằng số mũ bits trên)

- Số bit để đánh số hiệu trang vật lý là: 8 = 23, vậy n2 = 3.

- Số bit để đánh địa chỉ vật lý là: 32 = 25, vậy n = 5 (n = n2 + n1).

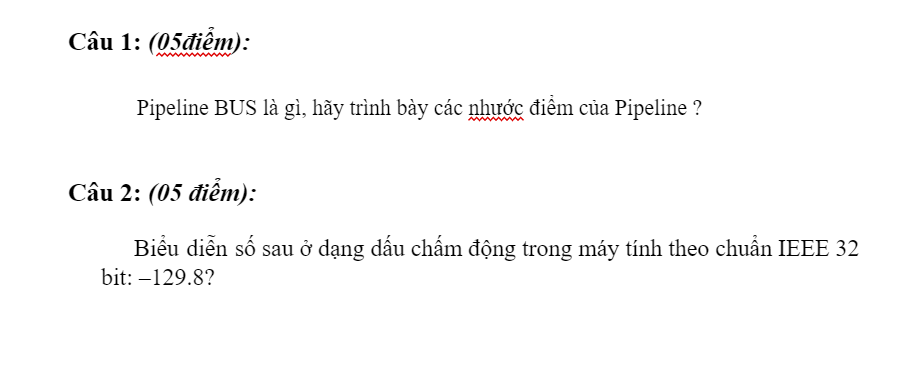
- Số trang logic của chương trình = dung lượng chương trình/ L = 16/4 = 4. Vậy số trang logic của chương trình phải là 4. (bảng PCB trên là phù hợp).

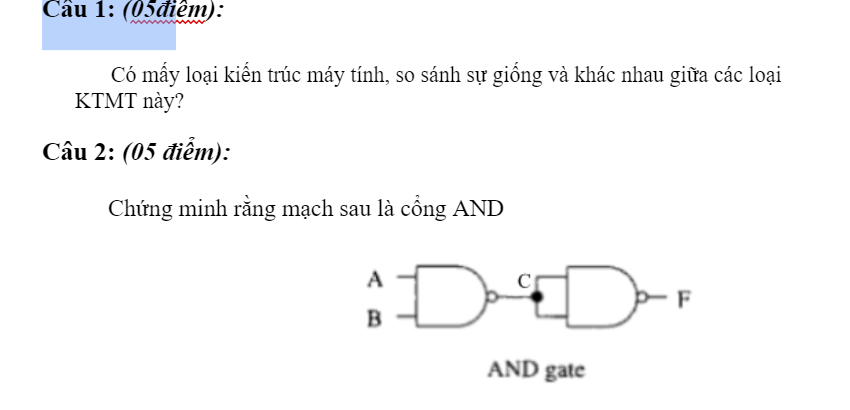
- Căn cứ vào bảng PCB, dung lượng bộ nhớ vật lý và kích thước của một trang, ta có sơ đồ nạp trang như sau:



- Số hiệu trang logic cần truy nhập là: Np = 2 < 4 (số trang logic của chương trình) . nên số hiệu trang cần truy nhập là hợp lệ.

- Địa chỉ lệch cần truy nhập trong trang là: d = 4 = 4 (độ dài trang), nên địa chỉ lệch cần truy nhập trong trang là không hợp lệ. Do vậy tiến trình phải dừng.





Câu 1:

|  |  |
| --- | --- |
| Von Neumann | Non Von Neumann |
| -Được xây dựng theo ý tưởng “chương trình được lưu trữ”  -Đặc điểm:  +Dữ liệu vào chương trình được chứa trong bộ lọc ghi  +Bộ nhớ được đánh địa chỉ cho các ngăn nhớ không phụ thuộc vào nội dung của chúng  +Các lệnh của máy tính được thực hiện một cách tuần tự  -Gồm 5 thành phần chính:  +Bộ xử lý trung tâm(CPU)  +Bộ nhớ làm việc(RAM)  +Bộ nhớ vĩnh viễn(ROM)  +Thiết bị vào(Input)  +Thiết bị ra(Output) | -Thuật ngữ Non Von Neumann thường được dành riêng cho các máy đại diện cho sự ra đi triệt để khỏi mô hình Von Neumann và do đó thường không được áp dụng cho đa bộ xử lý hoặc đa máy tính.  -Một máy không Von Neumann có thể không có khái niệm về lường điều khiển tuần tự(Nghĩa là không có bất kỳ thanh ghi nào tương ứng với 1 chương trình đối lập và không chỉ ra điểm hiện tại đã đạt được khi thực hiện chương trình)  -Không có khái niệm về 1 biến(tức là không có các vị trí lưu trữ có tên và sau đó được tham chiếu hoặc thay đổi) |

Câu 2:

C1: Ta có: F=NAND(C,C)=NAND(NAND(A,B),NAND(A,B))=   A.B.A.B = A.B=A.B(dpcm)

C2:

Ta có bảng chân lý của mạch AND

|  |  |  |
| --- | --- | --- |
| A | B | K |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

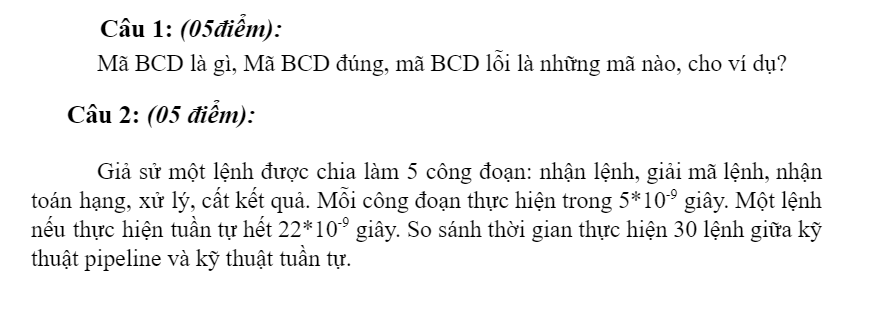
Ta thấy:

C=NAND(A,B)

F=NAND(C,C)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | C | F | K |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 |

🡺Đầu ra của mạch trùng với bảng chân lý của mạch AND🡺 Mạch trên là cổng AND



M = 5

Tc = 5 \*10^-9

T = 22 \* 10^-9

N = 30

Ts = m \* T = 5 \* 22 \* 10^-9

Tp = m \* Tc + (n-1) \* Tc = 5\*5\*10^-9 + 29 \* 5 \* 10^-9

* Ts – Tp =

Cho câu lệnh gồm 5 công đoạn: IF, ID, OF, EX, WO. Thời gian thực hiện của mỗi công đoạn tương ứng là: 8ns, 4ns, 9ns, 6ns, 8ns. So sánh thời gian thực hiện 30 lệnh tuần tự và đường ống.

**T =** 8ns+ 4ns+ 9ns+ 6ns+ 8ns = 35ns

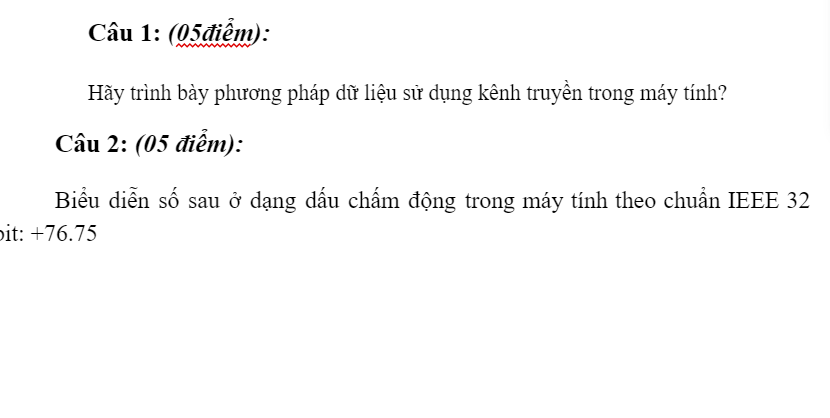
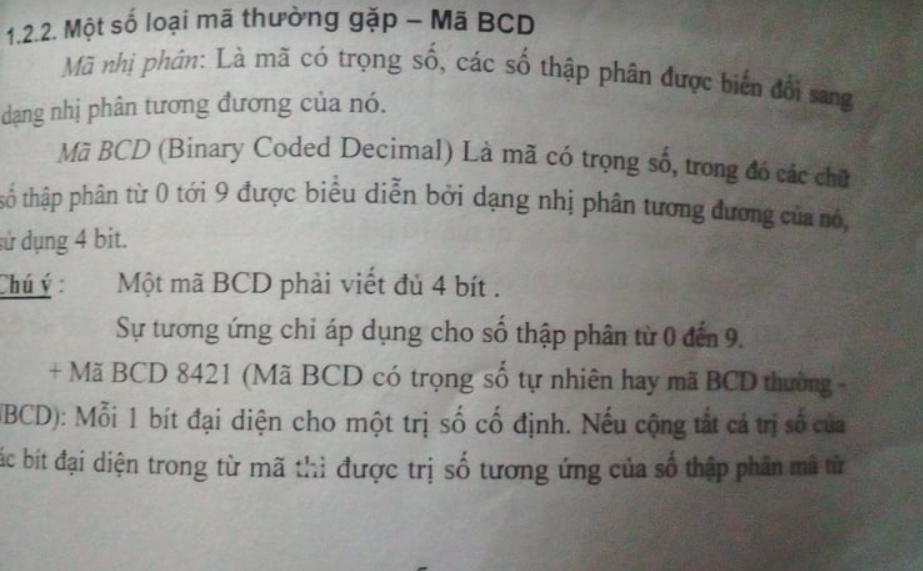
Tc = 9ns

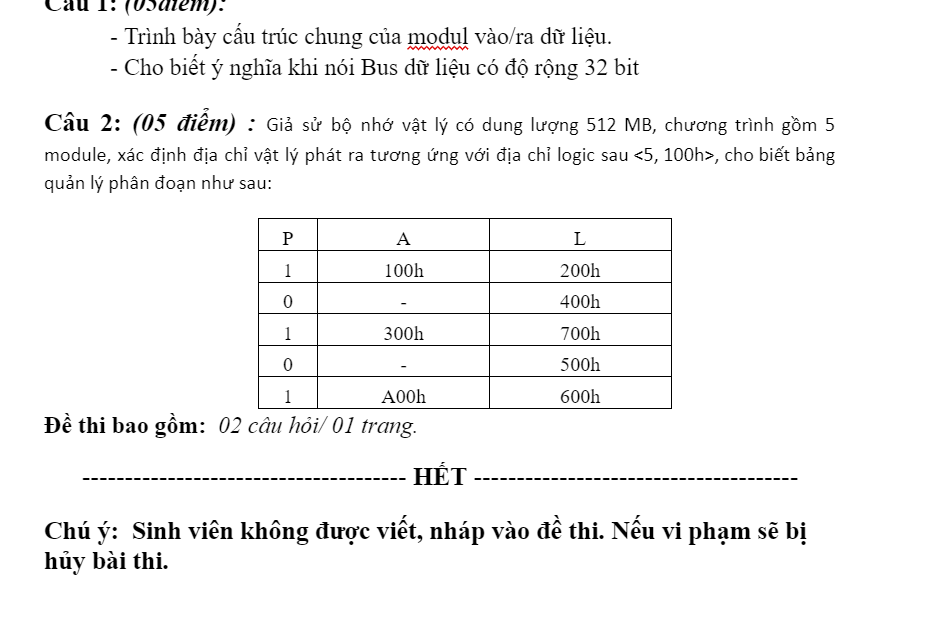
N = 30

Ts = m \* T = 5 \* 35 \* 10^-9

Tp = m \* Tc + (n-1) \* Tc = 5\*9\*10^-9 + 29 \* 9 \* 10^-9

* Ts – Tp =

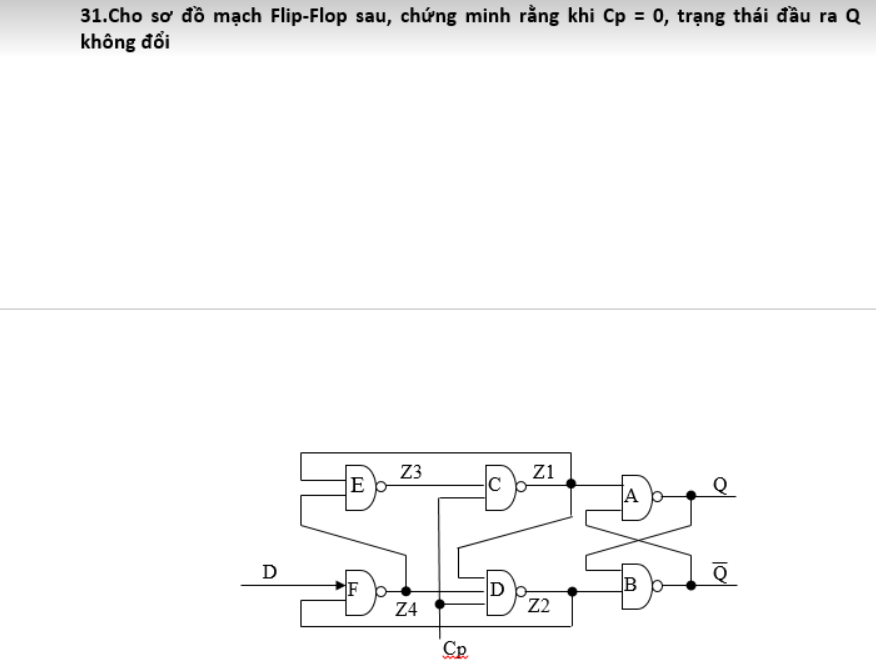




* P - trường dấu hiệu:
  + P = 0 : đoạn chưa được nạp,nằm ở vùng swap do hệ thống quản lý.
  + P = 1: đoạn đã được nạp vào bộ nhớ chính.
* A – trường địa chỉ: cho biết địa chỉ đầu vùng nhớ mà đoạn đã được nạp, trường này chỉ có ý nghĩa khi P = 1.
* L – trường độ dài: cho biết độ dài của đoạn, hay chính là dung lượng của module khi biên dịch.
* Địa chỉ truy nhập lệnh hoặc dữ liệu được biểu diễn là bộ đôi giá trị **<S, d>**, địa chỉ này được gọi là địa chỉ logic.
  + S là số hiệu của đoạn cần truy nhập.
  + d là địa chỉ tương đối của lệnh hay dữ liệu trong đoạn.
* Bảng SCB cho thấy chương trình có 5 đoạn, đánh số từ 0 đến 4.

- Số hiệu đoạn cần truy nhập là S =5 <= 5, địa chỉ đoạn cần truy nhập là không  hợp lệ, tiến trình dừng





Z2 = 1

Z1 = 1

+ D = 0

Z4 = NAND(D,Z2) = NAND(0,1) = 1