**ÔN TẬP**

**A. LÝ THUYẾT**

1. Thanh ghi của vi xử lý là gì? Nêu chức năng và đặc điểm của thanh ghi tích luỹ A. Nêu chức năng và phương thức hoạt động của bộ đếm chương trình PC (thanh ghi lệnh IP).

**a. Thanh ghi của vi xử lý là thành phần nhớ bên trong CPU:**

+ Lưu trữ tạm thời lệnh và dữ liệu cho CPU xử lý;

+ Có dung lượng nhỏ, số lượng ít;

+ Tốc độ rất cao (bằng tốc độ CPU).

**Chức năng và đặc điểm của thanh ghi tích luỹ A:**

+ Lưu trữ các toán hạng đầu vào và lưu kết quả đầu ra. Cũng được sử dụng để trao đổi dữ liệu với các thiết bị vào.

+ Kích thước của thanh ghi A tương ứng với đồ dài của từ dữ liệu CPU: 8, 16, 32, 64 bit

**b. Chức năng của bộ đếm chương trình PC:**

Luôn chứa địa chỉ của ô nhớ chứa lệnh kế tiếp được thực hiện.

**Phương thức hoạt động của bộ đếm chương trình PC:**

+ PC chứa địa chỉ ô nhớ chứa lệnh đầu tiên của chương trình được kích hoạt và được Hệ điều hành tải vào bộ nhớ.

+ Khi CPU thực hiện xong một lệnh, địa chỉ của ô nhớ chứa lệnh tiếp theo được nạp vào PC

**2 Trình bày khái niệm và vai trò của bộ nhớ cache. Giải thích hai nguyên lý hoạt động của cache. Cho ví dụ minh hoạ cho hai nguyên lý trên.**

**Khái niệm và vai trò của bộ nhớ cache:**

+ Bộ nhớ Cache là một thành phần của cấu trúc phân cấp của hệ thống bộ nhớ Cache đóng vai trong trung gian, trung chuyển dữ liệu từ bộ nhớ chính về CPU và ngược lại.

+ Bộ nhớ cache giúp tăng hiệu năng hệ thống nâng cao tốc độ và băng thông trao đổi dữ liệu giữa CPU và cache. Đồng thời, Cache giúp giảm giá thành sản xuất.

**Nguyên lý cục bộ/ lân cận về không gian (spatial locality) và Ví dụ:**

+ Nếu một vị trí bộ nhớ được truy cập, thì xác suất các vị trí gần đó được truy cập trong thời gian gần tới là cao.

+ Áp dụng với dữ liệu và các lệnh có thứ tự tuần tự theo chương trình.

**Ví dụ:** Các lệnh liên tiếp trong một chương trình hoặc các ví dụ mở tuần tự các ảnh trong cùng một album ảnh trên máy tính..vv

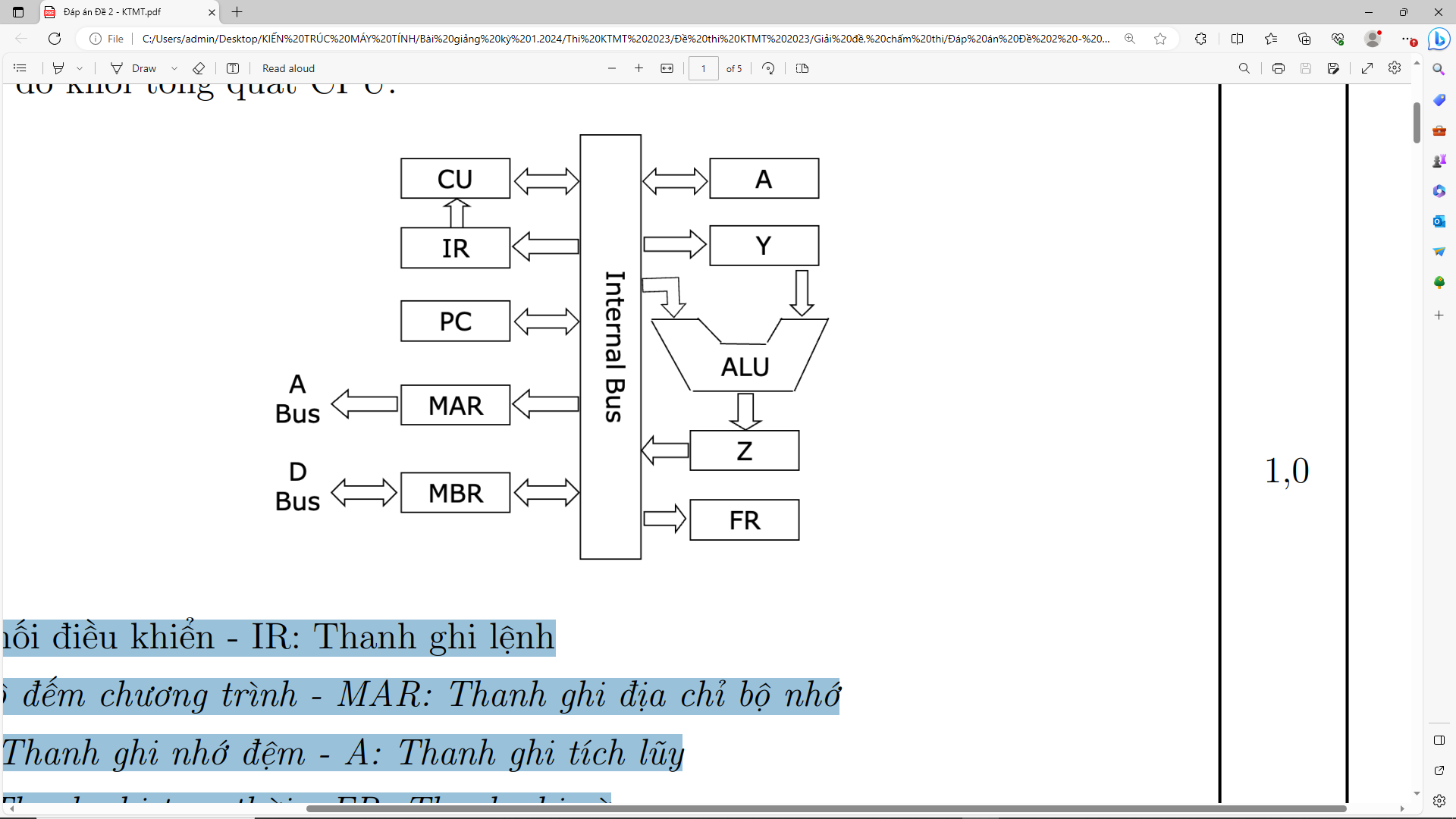
**Nguyên lý cục bộ/ lân cận về thời gian (temporal locality) và Ví dụ:**

+ Nếu một vị trí bộ nhớ được truy cập, thì xác suất nó được truy cập lại trong thời gian gần là cao.

+ Áp dụng với các mục dữ liệu và các lệnh trong vòng lặp.

**Ví dụ:** Các lệnh trong vòng lặp..vv.

**3. Vẽ sơ đồ khối tổng quát CPU và trình bày chu trình xử lý lệnh của CPU.**



CU: Khối điều khiển - IR: Thanh ghi lệnh

PC: Bộ đếm chương trình - MAR: Thanh ghi địa chỉ bộ nhớ

MBR: Thanh ghi nhớ đệm - A: Thanh ghi tích lũy

Y, Z: Thanh ghi tạm thời - FR: Thanh ghi cờ

ALU: Khối tính toán số học-logic

**Chu trình xử lý lệnh của CPU:**

1. Khi chương trình chạy, hệ điều hành tải mã chương trình vào bộ nhớ RAM

2. Địa chỉ lệnh đầu tiên của chương trình được đưa vào thanh ghi PC

3. Địa chỉ của ô nhớ chứa lệnh được chuyển tới bus A qua thanh ghi MAR

4. Bus A truyền địa chỉ tới khối quản lý bộ nhớ MMU (Memory Management Unit) và chọn ô nhớ, sinh ra tín hiệu READ

5. Lệnh chứa trong ô nhớ được chuyển tới thanh ghi MBR qua bus D

6. MBR chuyển lệnh tới thanh ghi IR. Sau đó IR lại chuyển lệnh tới CU

7. CU giải mã lệnh và sinh ra các tín hiệu xử lý cho các đơn vị khác

8. Địa chỉ trong PC được tăng lên để trỏ tới lệnh tiếp theo sẽ được thực hiện

9. Thực hiện lại các bước 3–>9 để chạy hết các lệnh của chương t

**4. Cơ chế ống lệnh (pipeline) của CPU thường gặp phải những vấn đề gì? Nêu ý nghĩa các lệnh dưới đây (R1, R2 là các thanh ghi và các lệnh quy ước theo dạng LỆNH <ĐÍCH> <GỐC>) và đưa ra một hướng giải quyết xung đột dữ liệu trong pipeline khi thực hiện đoạn chương trình sau:**

(1) LOAD R2, #200

(2) LOAD R1, #1000

(3) STORE (R1), R2

(4) SUBSTRACT R2, #210

(5) ADD 1000, #10

(6) ADD R2, (R1)

biết mỗi lệnh được chia thành 5 giai đoạn trong pipeline: Đọc lệnh (IF), giải mã đọc toán hạng (ID), truy nhập bộ nhớ (MEM), thực hiện (EX) và lưu kết quả (WB).

**Ý nghĩa của các lệnh chương trình:**

(1) LOAD R2, #200; Gán R2 = 200

(2) LOAD R1, #1000; Gán R1 = 1000

(3) STORE (R1), R2; Lưu R2 = 200 vào ô nhớ M[R1] hay ô nhớ M[1000] = 200

(4) SUBSTRACT R2, #210; R2 = R2 - 210 = 200 - 210 = -20

(5) ADD 1000, #10; M[1000] = M[1000] + 10 = 200 + 10 = 210

(6) ADD R2, (R1); R2 = R2 + M[R1] = R2 + M[1000] = -20 + 210 = 190

**Vấn đề trong đoạn lệnh trên là xung đột, tranh chấp dữ liệu Data Hazard:**

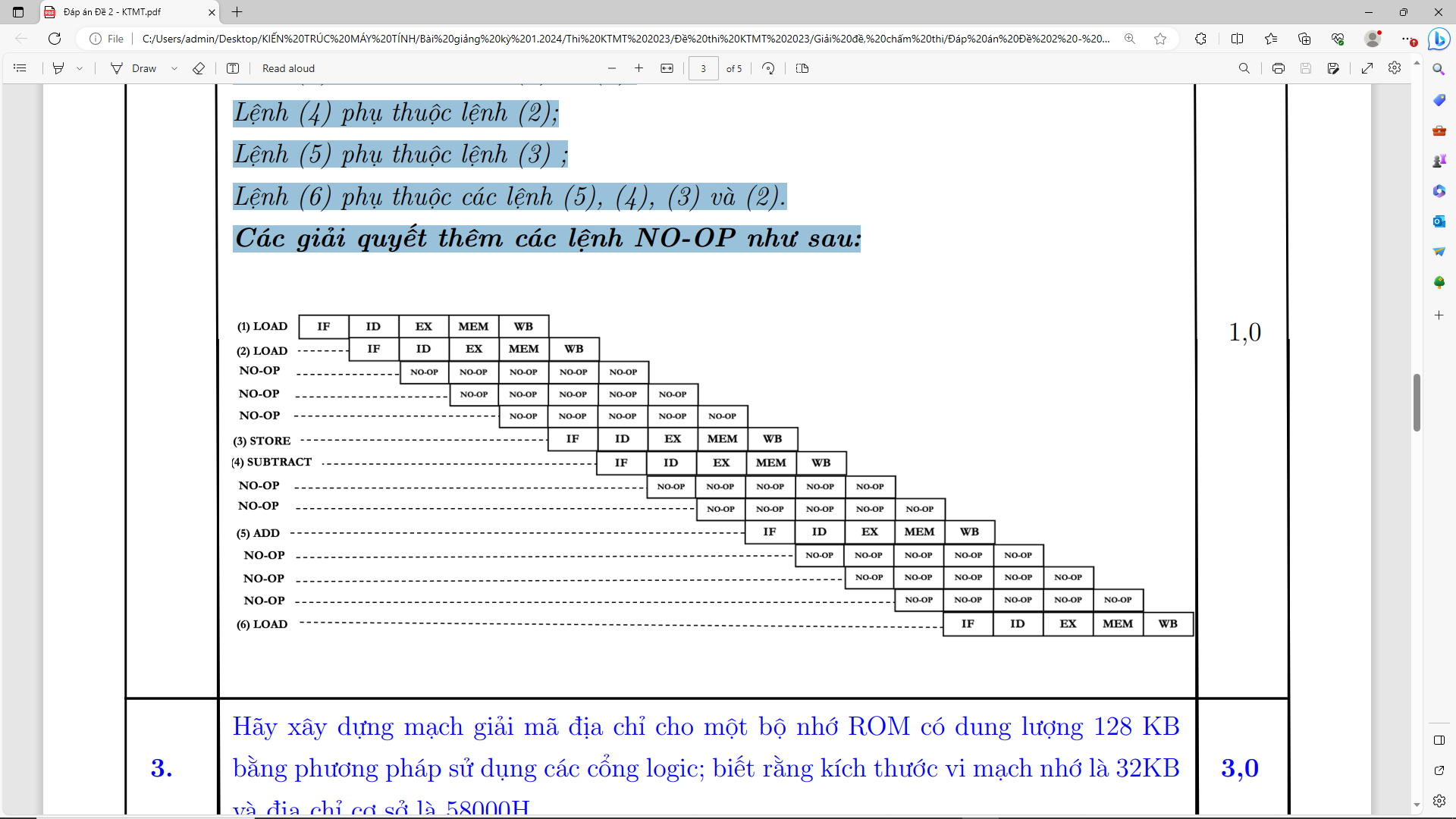
Lệnh (3) phụ thuộc lệnh (1) và (2);

Lệnh (4) phụ thuộc lệnh (2);

Lệnh (5) phụ thuộc lệnh (3) ;

Lệnh (6) phụ thuộc các lệnh (5), (4), (3) và (2).

**Các giải quyết thêm các lệnh NO-OP như sau:**



**5. Cho biết vai trò của vào ra dữ liệu, có các phương pháp nào để vào ra dữ liệu. Phương pháp vào ra bằng thăm dò là gì? Cho ví dụ minh hoạ?**

**Vai trò của vào ra dữ liệu:**

+ Là phương tiện giúp CPU giao tiếp với thế giới bên ngoài

+ Cung cấp dữ liệu đầu vào cho CPU xử lý

+ Cung cấp phương tiện để CPU kết xuất dữ liệu đầu ra

\*Các phương pháp vào ra chính:

+ Thăm dò (polling)

+ Ngắt (Interrupt)

+ Truy nhập trực tiếp bộ nhớ (DMA-Direct Memory Access)

**Phương pháp vào ra bằng thăm dò:**

+ CPU quản lý danh sách các thiết bị vào ra kèm theo địa chỉ các cổng giao tiếp;

+ Các thiết bị vào ra định kỳ cập nhật trạng thái sẵn sàng làm việc của mình lên

các bít cờ trạng thái vào ra của mình;

+ CPU định kỳ lần lượt quét các thiết bị vào ra để đọc bit cờ trạng thái vào ra;

- Nếu gặp một thiết bị sẵn sàng làm việc, 2 bên tiến hành trao đổi dữ liệu;

- Trao đổi dữ liệu xong, CPU tiếp tục quét thiết bị khác.

+ CPU là bên chủ động trong quá trình trao đổi dữ liệu

**Lấy ví dụ phương pháp vào ra bằng thăm dò:**

**6. Nêu các dạng địa chỉ của lệnh trong Vi xử lý 8086. Cho một ví dụ minh hoạ với mỗi dạng địa chỉ đó.**

**Các dạng địa chỉ của lệnh trong Vi xử lý 8086:**

• Toán hạng 2 địa chỉ

– Khuôn dạng: OPCODE Addr1, Addr2

– Mỗi địa chỉ Addr1, Addr2 cùng tham chiếu đến ô nhớ hoặc thanh ghi.

– Ví dụ: ADD AH, BL

• Toán hạng 1 địa chỉ

– Khuôn dạng: OPCODE Addr

– Địa chỉ Addr tham chiếu đến một ô nhớ hoặc thanh ghi.

– Ví dụ: MUL BL

• Toán hạng 1,5 địa chỉ

– Khuôn dạng: OPCODE Addr1, Addr2

– Một địa chỉ tham chiếu tới 1 ô nhớ và địa chỉ còn lại tham chiếu tới 1 thanh ghi. Là dạng hỗn hợp giữa các toán hạng thanh ghi và vị trí bộ nhớ.

– Ví dụ: MOV AL, (BX)

• Toán hạng 0 địa chỉ

– Khuôn dạng: OPCODE

– Lệnh không có địa chỉ toán hạng

– Ví dụ: CLC; NOP; RET..vv.

**7. Thế nào là vào ra bằng thăm dò và vào ra bằng truy nhập trực tiếp bộ nhớ (DMA). So sánh sự giống, khác nhau của hai phương pháp và lấy ví dụ minh hoạ.**

**Vào ra bằng thăm dò:**

+ CPU quản lý danh sách các thiết bị vào ra kèm theo địa chỉ các cổng giao tiếp;

+ Các thiết bị vào ra định kỳ cập nhật trạng thái sẵn sàng làm việc của mình lên các bít cờ trạng thái vào ra của mình;

+ CPU định kỳ lần lượt quét các thiết bị vào ra để đọc bit cờ trạng thái vào ra;

- Nếu gặp một thiết bị sẵn sàng làm việc, 2 bên tiến hành trao đổi dữ liệu;

- Trao đổi dữ liệu xong, CPU tiếp tục quét thiết bị khác.

+ CPU là bên chủ động trong quá trình trao đổi dữ liệu

**Vào ra bằng truy cập trực tiếp bộ nhớ DMA:**

+ Phương pháp vào ra bằng DMA (Direct Memory Access) cho phép thiết bị vào ra trao đổi dữ liệu trực tiếp với bộ nhớ theo khối, không thông qua CPU;

+ DMA thích hợp khi cần trao đổi dữ liệu với khối lượng lớn trong khoảng thời gian ngắn.

+ Để vào ra bằng DMA cần dùng thêm mạch chuyên dụng DMAC để điều khiển truy cập trực tiếp vào bộ nhớ.

**So sánh giống và khác nhau giữa vào ra bằng thăm dò và vào ra bằng truy nhập trực tiếp bộ nhớ:**

• Giống nhau:

- Đều có mục đích cho phép trao đổi dữ liệu giữa máy tính và các thiết bị ngoại vi.

• Khác nhau:

|  |  |
| --- | --- |
| **Vào ra bằng thăm dò** | **Truy nhập trực tiếp DMA** |
| + CPU quản lý và là bên chủ động trong quá trình trao đổi dữ liệu | + DMA can thiệp vào quá trình trao đổi dữ liệu không thông qua CPU |
| + CPU phải thăm dò từng thiết bị nên hiệu quả thấp và không khả thi có nhiều thiết bị cần thăm dò | + Hiệu suất và tốc độ cao hơn nhiều lần so với các phương pháp khác |
| + Đơn giản dễ cài đặt, có thể cài đặt bằng phần mềm | + Phức tạp do cần dùng phần cứng để điều khiển quá trình DMA |

**8. Nêu chức năng và phương thức hoạt động của con trỏ ngăn xếp SP (Stack Pointer). Thanh ghi cờ (hay thanh ghi trạng thái) của vi xử lý có chức năng gì? Nêu ý nghĩa của các cờ nhớ (C), cờ không (Z), cờ dấu (S).**

**Chức năng của con trỏ ngăn xếp SP (Stack Pointer):** là một thanh ghi luôn chứa địa chỉ đỉnh ngăn xếp (Stack). Có hai thao tác chính với ngăn xếp POP và PUSH.

**Phương thức hoạt động của con trỏ ngăn xếp SP (Stack Pointer):** Có hai hoạt động của ngăn xếp đó là hoạt động PUSH và hoạt động POP.

• Push - đẩy dữ liệu vào ngăn xếp:

– SP <— SP + 1 ; tăng địa chỉ đỉnh ngăn xếp

– {SP} <— Dữ liệu ; nạp dữ liệu vào ngăn xếp

• Pop - lấy dữ liệu ra khỏi ngăn xếp:

– Thanh ghi <— {SP} ; chuyển dữ liệu từ đỉnh ngăn xếp vào thanh ghi

– SP <— SP – 1 ; giảm địa chỉ đỉnh ngăn xếp.

**Thanh ghi cờ (hay thanh ghi trạng thái) của vi xử lý có chức năng gì? Nêu ý nghĩa của các cờ nhớ (C), cờ không (Z), cờ dấu (S).**

**Chức năng của thanh ghi cờ của vi xử lý:** là một thanh ghi đặc biệt của CPU: mỗi bit của thanh ghi cờ lưu trạng thái của kết quả của phép tính ALU thực hiện.

**Ý nghĩa của các cờ nhớ (C), cờ không (Z), cờ dấu (S):**

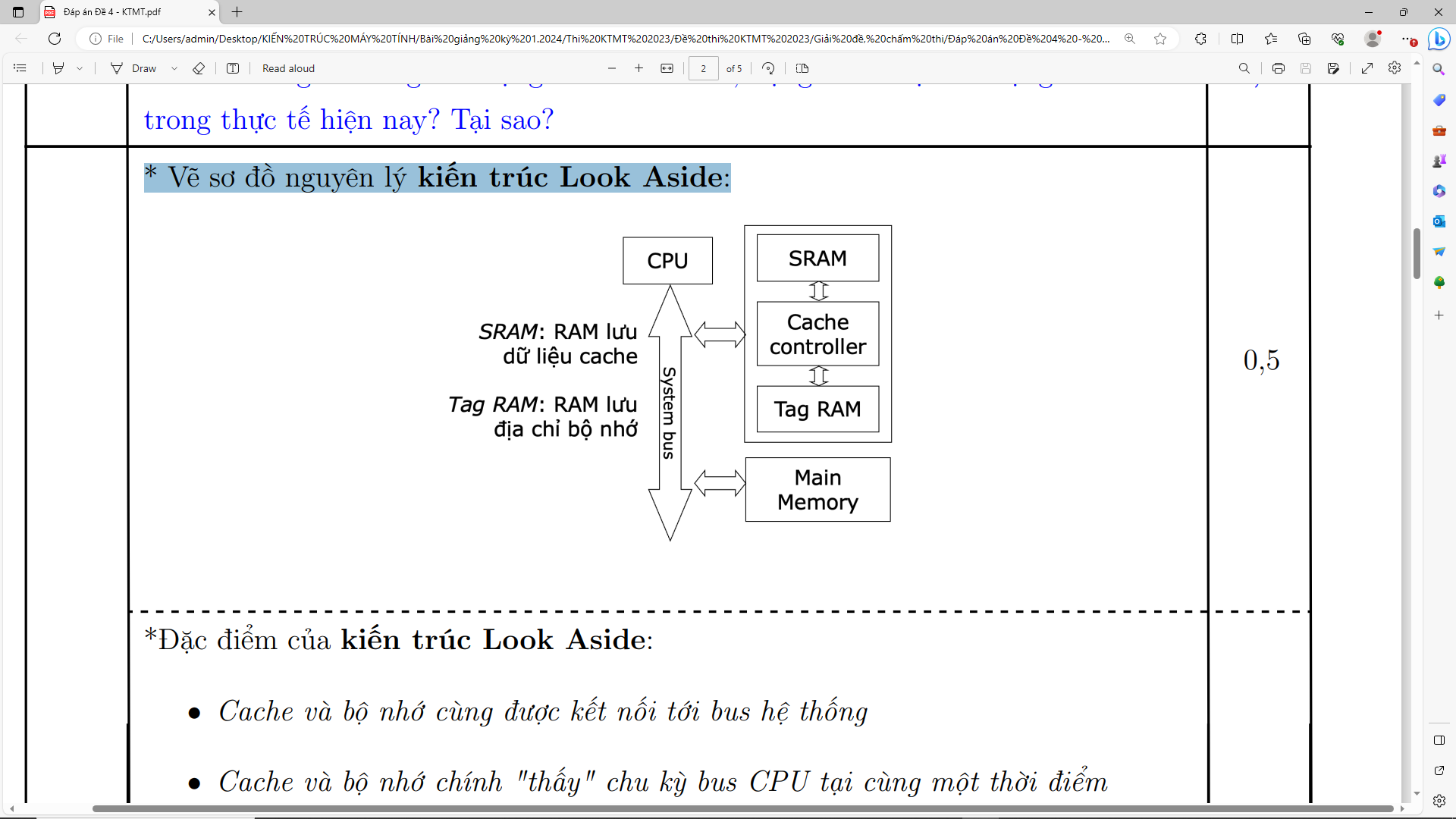
+ CF: Cờ nhớ, CF = 1 nếu có nhớ/mượn, CF = 0 trong trường hợp khác.

+ ZF: Cờ Zero, ZF = 1 nếu kết quả = 0 và ZF = 0 nếu kết quả khác 0.

+ SF: Cờ dấu, SF = 1 nếu kết quả âm và SF = 0 nếu kết quả dương.

**9. Vẽ sơ đồ nguyên lý và nêu đặc điểm của hai dạng kiến trúc cache: Look Aside và Look Through. Trong hai dạng kiến trúc trên, dạng nào được sử dụng nhiều hơn trong thực tế hiện nay? Tại sao?**

**Vẽ sơ đồ nguyên lý kiến trúc Look Aside:**



**Đặc điểm của kiến trúc Look Aside:**

• Cache và bộ nhớ cùng được kết nối tới bus hệ thống

• Cache và bộ nhớ chính "thấy" chu kỳ bus CPU tại cùng một thời điểm

• Ưu điểm:

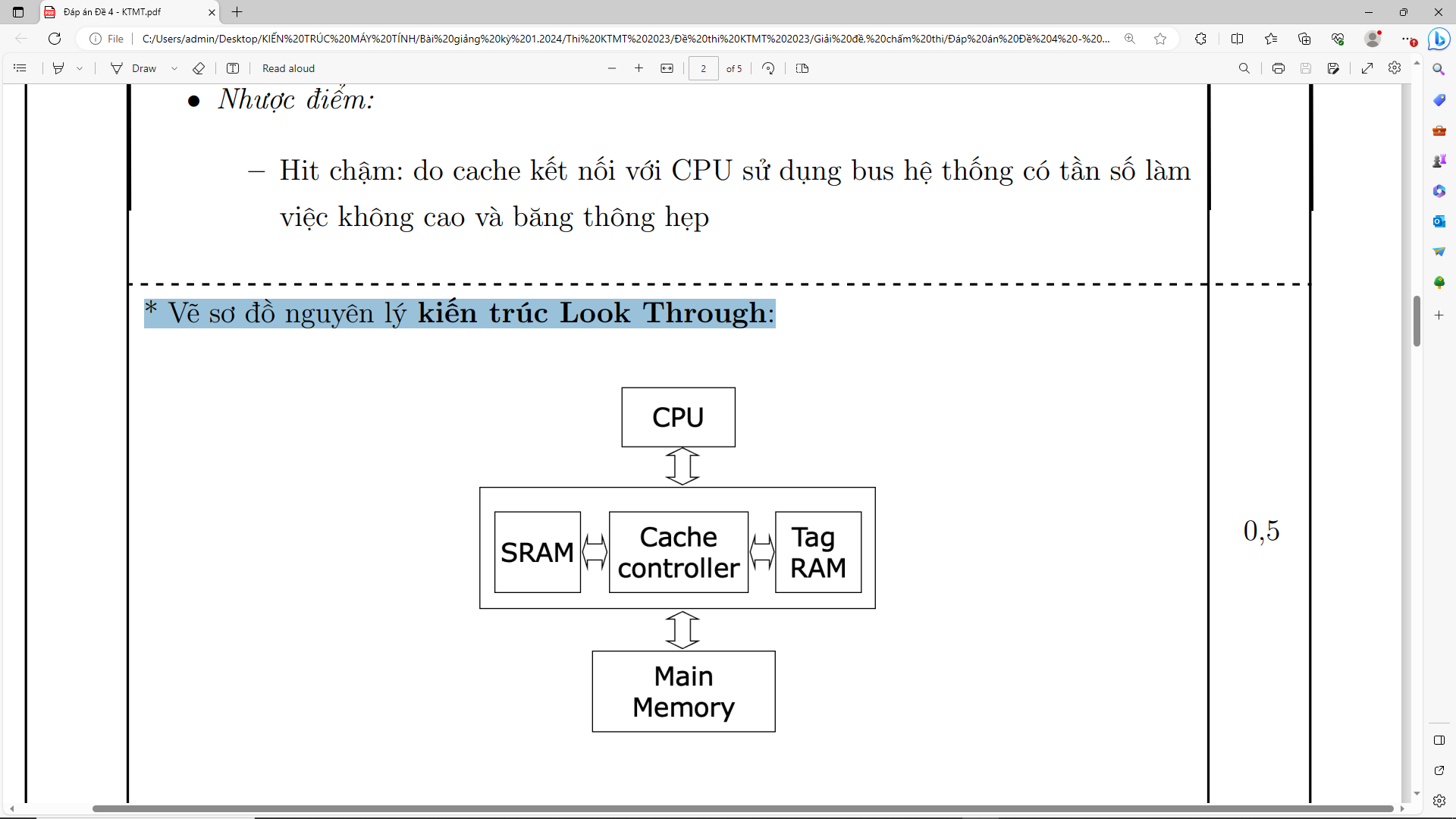
– Thiết kế đơn giản

– Miss nhanh: khi CPU không tìm thấy mục dữ liệu trong cache nó đồng thời tìm trong bộ nhớ chính cùng 1 chu kỳ xung nhịp.

• Nhược điểm:

– Hit chậm: do cache kết nối với CPU sử dụng bus hệ thống có tần số làm việc không cao và băng thông hẹp

**Vẽ sơ đồ nguyên lý kiến trúc Look Through:**



**Đặc điểm của kiến trúc Look Through:**

• Cache nằm giữa CPU và bộ nhớ chính

• Cache "thấy" chu kỳ bus CPU trước, sau đó nó "truyền" lại cho bộ nhớ chính.

• Ưu điểm:

– Hit nhanh: Cache kết nối với CPU bằng bus riêng tốc độ cao và có bang thông lớn

• Nhược điểm:

– Đắt

– Thiết kế phức tạp

– Miss chậm: khi CPU không tìm thấy dữ liệu trong cache nó cần tìm trong Bộ nhớ tại 1 xung nhịp tiếp theo

**Kiến trúc nào được sử dụng nhiều hơn trong thực tế hiện nay? Tại sao?**

Trong hai kiến trúc trên, thực tế kiến trúc dạng Look Through được sử dụng nhiều hơn vì ưu điểm của kiến trúc Look Through là tỷ lệ hit nhanh hơn do Cache được kết nối với CPU bằng bus riêng ngược lại Look Aside hit chận hơn.

**B. BÀI TẬP**

**I. BỘ NHỚ CACHE**

**1. Máy tính có dung lượng bộ nhớ chính: 1MB, kích thước của một dòng cache 16 bytes, bộ nhớ cache có kích thước 64KB. Xác định số bit trong các thành phần địa chỉ của ô nhớ và tính kích thước bộ nhớ cache với phương pháp ánh xạ:**

a. Ánh xạ trực tiếp (direct-mapped cache)

b. Ánh xạ tập kết hợp 4 đường (4-way set-associative cache).

**Bài giải**

**a. Ánh xạ trực tiếp (direct-mapped cache):**

***Xác định số bit trong các thành phần địa chỉ của ô nhớ :***

MMemory = 1MB = 220B ⇒ Tag+Line+Word = 20bit

MCache = 64KB = 216B. [Line] = 16byte = 24B

• Số trang nhớ của bộ nhớ:

P =MMemory/MCache = 220B/216B = 24 ⇒ Tag = 4bit

• Số dòng (line): L =MCache/[Word] = 216B/24B = 212 ⇒ Line = 12bit

• Word = 20 - Tag - Line = 4 ⇒ Tag = 4 ; Line = 12 ; Word = 4.

***Tổng số bit cần thiết cho bộ nhớ Cache:***

+ Số lượng bit cần thiết cho một dòng Cache là:

bits/block = data bits + tag bits + valid bit = 20 + 4 + 1 = 25bit

+ Tổng số bit cần thiết cho bộ nhớ Cache là:

L × bits/block = 212 x 25 bit= 12,5 Kbytes

***b. Ánh xạ tập kết hợp 4 đường (4-way set-associative cache).***

**Xác định số bit trong các thành phần địa chỉ của ô nhớ:**

MMemory = 1MB = 220B ⇒ Tag+Line+Word = 20bit

MCache = 64KB = 216B. [Line] = 16byte = 24B 🡪 [Way] = 4 = 22

• Số trang nhớ của bộ nhớ:

P = (MMemory×[Way])/MCache= (220B×22)/216B = 26 ⇒ Tag = 6bit

• Số dòng (line):

L = MCache/[Word]×[Way] = 216B/24B×22 = 210 ⇒ Line = 10bit

• Word = 20 - Tag - Line = 4 ⇒ Tag = 6 ; Line = 10 ; Word = 4.

**Tổng số bit cần thiết cho bộ nhớ Cache:**

+ Số lượng bit cần thiết cho một dòng Cache là:

bits/block = data bits + tag bits + valid bit = 20 + 6 + 1 = 27bit

+ Tổng số bit cần thiết cho bộ nhớ Cache là:

Way × L× bits/block = 22 × 210 × 27 bit= 13,5 Kbytes

**2. Tổng số bit của cache**

**Tính tổng số bit cần thiết để thực hiện một cache theo kiểu direct mapped có thể chứa 128 KB dữ liệu với kích cỡ block là 1 word**.

• Dữ liệu của cache = 128 KB = 217 bytes = 215 words = 215 blocks

• Số bit để thực hiện 1 block = Số bit dữ liệu + Số bit tag + Bit Valid

= 32 + (32 – 15 – 2) + 1 = 48 bits

• Tổng số bit cần thiết = 215 × 48 bits = 215 × (1.5 × 32) bits

= 1.5 × 220 bits = 1.5 Mbits

o Dữ liệu của cache = 128 KB × 8 = 1024 KB = 1 Mbits

o Tổng số bit cần thiết/Dữ liệu của cache = 1.5/1 = 1.5

**3. Tính toán địa chỉ block ở bộ nhớ được định địa chỉ theo byte**

**Giả sử một direct mapped cache có 64 block và kích cỡ của một block là 16 byte. Byte tại địa chỉ 1200 ở bộ nhớ sẽ tương ứng với block nào ở cache?**

- Do kích cỡ của một block = 16 byte cho nên:

Địa chỉ byte 1200 🡪 Địa chỉ block ở bộ nhớ = [1200/16] = 75

- Do cache có 64 block cho nên:

Địa chỉ block 75 ở bộ nhớ được tính ở trên

🡪 Địa chỉ block ở cache = (75 mod 64) = 11

**II. LẬP TRÌNH HỢP NGỮ**

**1. Viết chương trình hợp ngữ Assembly cho phép nhập một chuỗi các ký tự, việc nhập kết thúc khi nhấn ENTER**

.MODEL small

.STACK 100H

.DATA

SO DW 0 ;chua so nhap

.CODE

Program:

mov AX,@data ;Khoi tao DS

mov DS, AX

xor CX,CX ;xoa CX = 0

mov BX,0Ah ;cho gia tri BX = 10

LAP:

mov AH,1 ;cho nhan tu ban phim 1 ky tu

int 21h

cmp AL,0Dh ;ky tu vua nhan la ENTER?

jE THOAT ;dung thoat

sub AL,30h ;chuyen so dang ASCII thanh so

mov CL,AL ;tam cat so vua nhap vao thanh ghi CL

mov AX,SO ;loi gia tri vao truoc ra AX

mul BX ;nhan voi 10

add AX,CX ;cong gia tri vao truoc voi so vua vao

mov SO,AX ;cat lai vao bien SO

jmp LAP

THOAT:

mov AH,4Ch ;tro ve DOS

int 21h

END Program

**2. Viết chương trình hợp ngữ Assembly cho phép nhập một chuỗi các ký tự, việc nhập kết thúc khi nhấn #**

.MODEL small

.STACK 100H

.DATA

SO DW 0 ; se chua so nhap

.CODE

Program3:

mov AX,@data ; Khoi tao DS

mov DS, AX

xor CX,CX ; xoa CX = 0

mov BX,0Ah ; cho gia tri BX = 10

LAP:

mov AH,1 ; cho nhan tu ban phim 1 ky tu

int 21h

cmp AL,23h ;ky tu vua nhan la #?

jE THOAT ; dung thoat

sub AL,30h ; chuyen so dang ASCII thanh so

mov CL,AL ; tam cat so vua nhap vao thanh ghi CL

mov AX,SO ; loi gia tri vao truoc ra AX

mul BX ; nhan voi 10

add AX,CX ; cong gia tri vao truoc voi so vua vao

mov SO,AX ; cat lai vao bien SO

jmp LAP

THOAT:

mov AH,4Ch ; tro ve DOS

int 21h

END Program3

**3.** **Viết chương trình hợp ngữ Assembly cho phép nhập một chuỗi các ký tự, việc nhập kết thúc khi nhấn # và yêu cầu in ra màn hình chuỗi ký tự đó theo thứ tự ngược lại.**

.Model small

.Stack 100H

.Data

STR\_IN DB 'Nhap\_vao\_mot\_chuoi\_ky\_tu\_ket\_thuc\_boi\_#:\_$'

STR\_OUT DB 10, 13, 'Xau\_ky\_tu\_dao\_nguoc\_la:\_$'

STR DB 100 DUP('$')

.code

Main Proc

MOV AX, @Data ; khoi tao thanh ghi DS

MOV DS, AX

MOV AH, 9

LEA DX, STR

MOV CX, 0

KIEM\_TRA:

MOV AH, 1

INT 21H

CMP AL, '#'

JE IN\_CHUOI

MOV [SI], AL

PUSH [SI]

INC SI

INC CX

JMP KIEM\_TRA

IN\_CHUOI:

MOV AH, 9

LEA DX, STR\_OUT

INT 21H

IN\_TUNG\_TY\_TU:

POP DX

MOV AH, 2

INT 21H

LOOP IN\_TUNG\_TY\_TU

MOV AH, 4CH

INT 21H

MAIN Endp

END MAIN

**4. Viết chương trình hợp ngữ Assembly 8086 để đếm số lần xuất hiện của chuỗi con "KTMT" trong một chuỗi và in kết quả ra màn hình dưới dạng số thập phân.**

.Model small

.Stack 100H

.Data

STR\_IN DB 100 DUP('$')

MSG DB 13, 10, 'So\_chuoi\_con\_KTMT\_la:\_$'

LAST DW 0

COUNT DW 0

BASE\_10 DW 10

.code

Main Proc

MOV AX, @Data ; khoi tao thanh ghi DS

MOV DS, AX

MOV AH, 0AH

LEA DX, STR\_IN

INT 21H

MOV AL, STR\_IN + 1

SUB AL, 4

MOV AH, 0

MOV LAST,AX

LEA SI, STR\_IN + 2

LEA DI, STR\_IN

MOV DX, 0

MOV CX, 0

TEST\_LOOP:

CMP [SI], '$'

JE DONE

CMP [SI], 'K'

JNE CONTINUE

CMP [SI+1], 'T'

JNE CONTINUE

CMP [SI+3], 'M'

JNE CONTINUE

CMP [SI], '$'

JE DONE

INC DX

CONTINUE:

INC SI

INC CX

JMP TEST\_LOOP

DONE:

MOV COUNT, DX

MOV AH, 09H

LEA DX, MSG

INT 21H

CALL OUTPUT

MOV AH, 4CH

INT 21H

MAIN Endp

OUTPUT PROC

MOV AX, COUNT

MOV CX, 0

DIVIDE:

MOV DX, 0

DIV BASE\_10

PUSH DX

INC CX

CMP AX, 0

JE SHOW

JMP DIVIDE

SHOW:

MOV AH, 02H

POP DX

ADD DL, '0'

INT 21H

DEC CX

CMP CX, 0

JNE SHOW

RET

OUTPUT ENDP

END MAIN

**5. Tính tổng đến 59. Lưu kết quả vào thanh ghi**

AX=0

BX=0

CX=59

AX++

BX=BX+AX

CX=0

THOÁT

**SAI**

**ĐÚNG**

.MODEL SMALL

.STACK 100h

.DATA

.CODE

MAIN PROC

MOV AX,0

MOV BX,0

MOV CX,59

TONG:

INC AX

ADD BX,AX

LOOP TONG

END MAIN

**6. Tính tổng đến 199. Xuất kết quả ra màn hình**

.MODEL SMALL

.STACK 100h

.DATA

S DW 0 ;khai bao bien S chua ket qua

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV AX,0

MOV CX,199

TONG:

ADD AX,CX

LOOP TONG

MOV S,AX

;HIEN THI KET QUA BANG CACH CHIA KET QUA CHO 10 LAY DU

MOV CX,0

MOV DX,0

MOV AX,S

MOV BX,10

CHIA:

MOV DX,0 ;DX:AX/BX

DIV BX

INC CX

PUSH DX

CMP AX,0

JNE CHIA

JMP XUAT

XUAT:

MOV AH,02H ;noi dung hien thi chua trong DX

POP DX

ADD DX,30H

INT 21H ;thuc thi ham 02h gui ma ascii hien thi len man hinh

LOOP XUAT

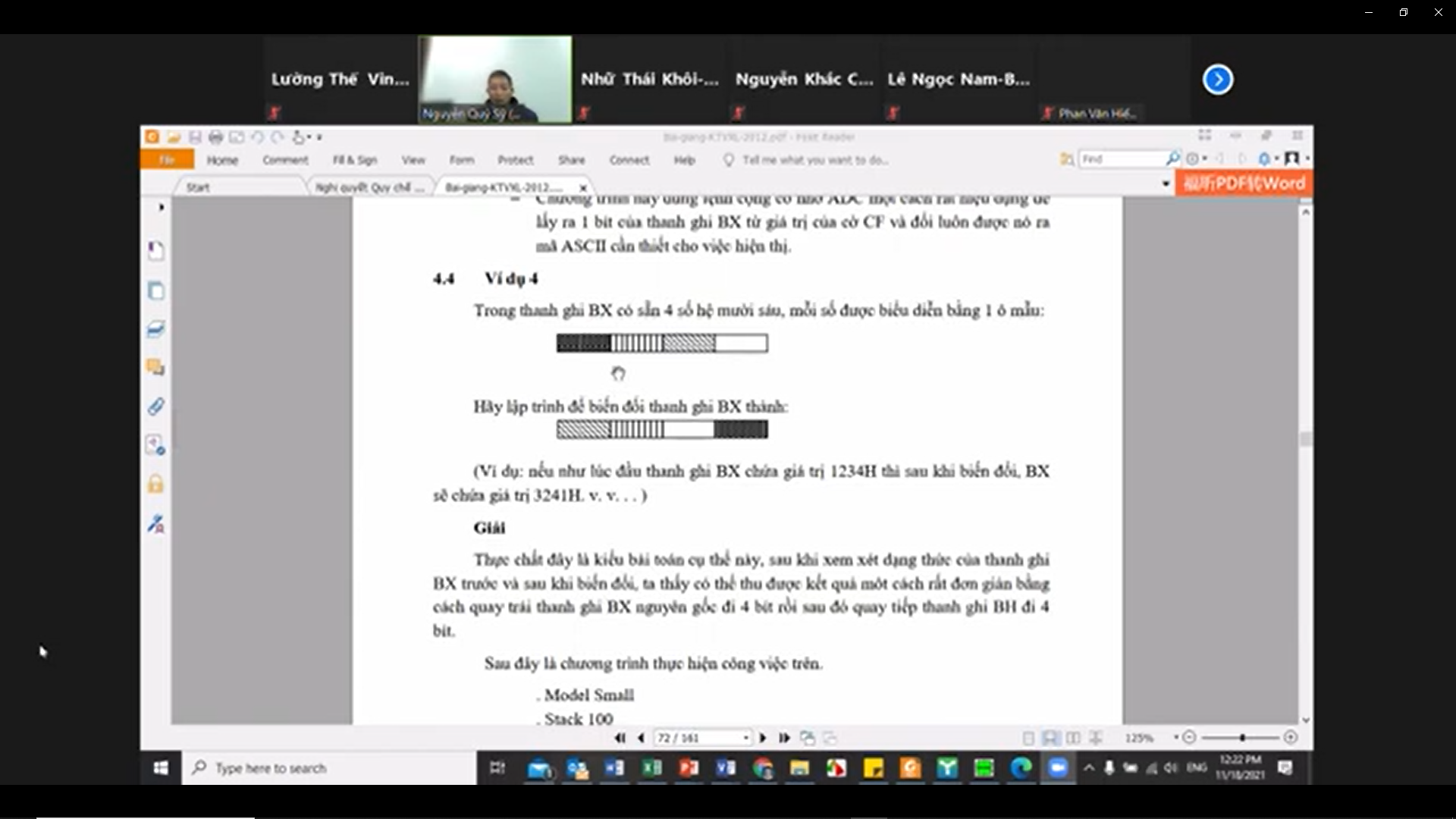
MOV AH,4CH

INT 21H

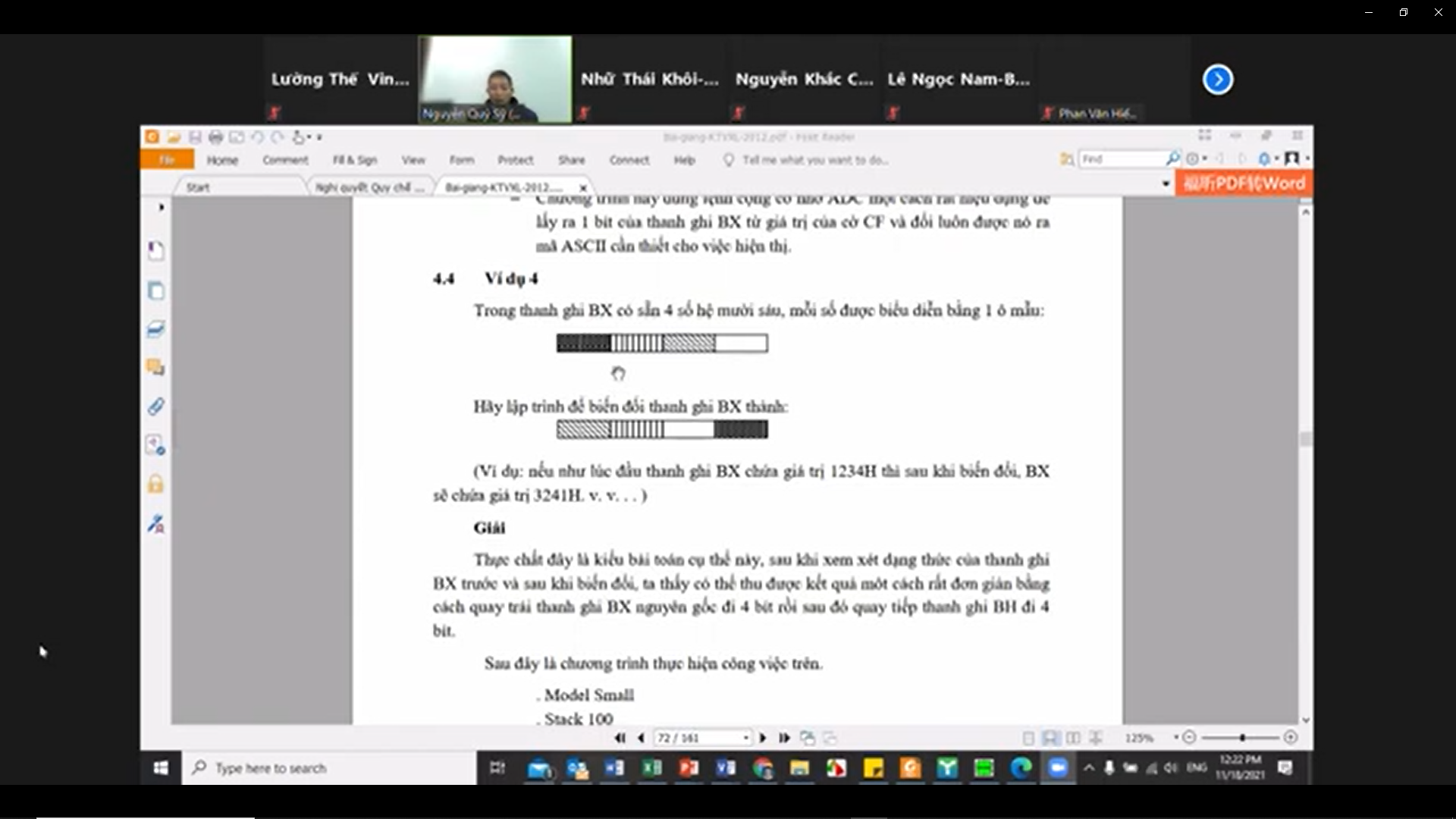
MAIN ENDP

END MAIN

**7. Thanh ghi BX có sẵn 4 số hệ 16, mỗi số được biểu diễn bằng 1 ô màu**



**Hãy lập trình để biển đổi thanh ghi thành**



.Model small

.Stack 100H

.Code

Main Proc

MOV CL,4

ROL BX, CL ;quay BX di 4 bit

MOV CL,4

ROR BH, CL ;trao 4 bit thap va cao cua BH

MOV AH,4CH

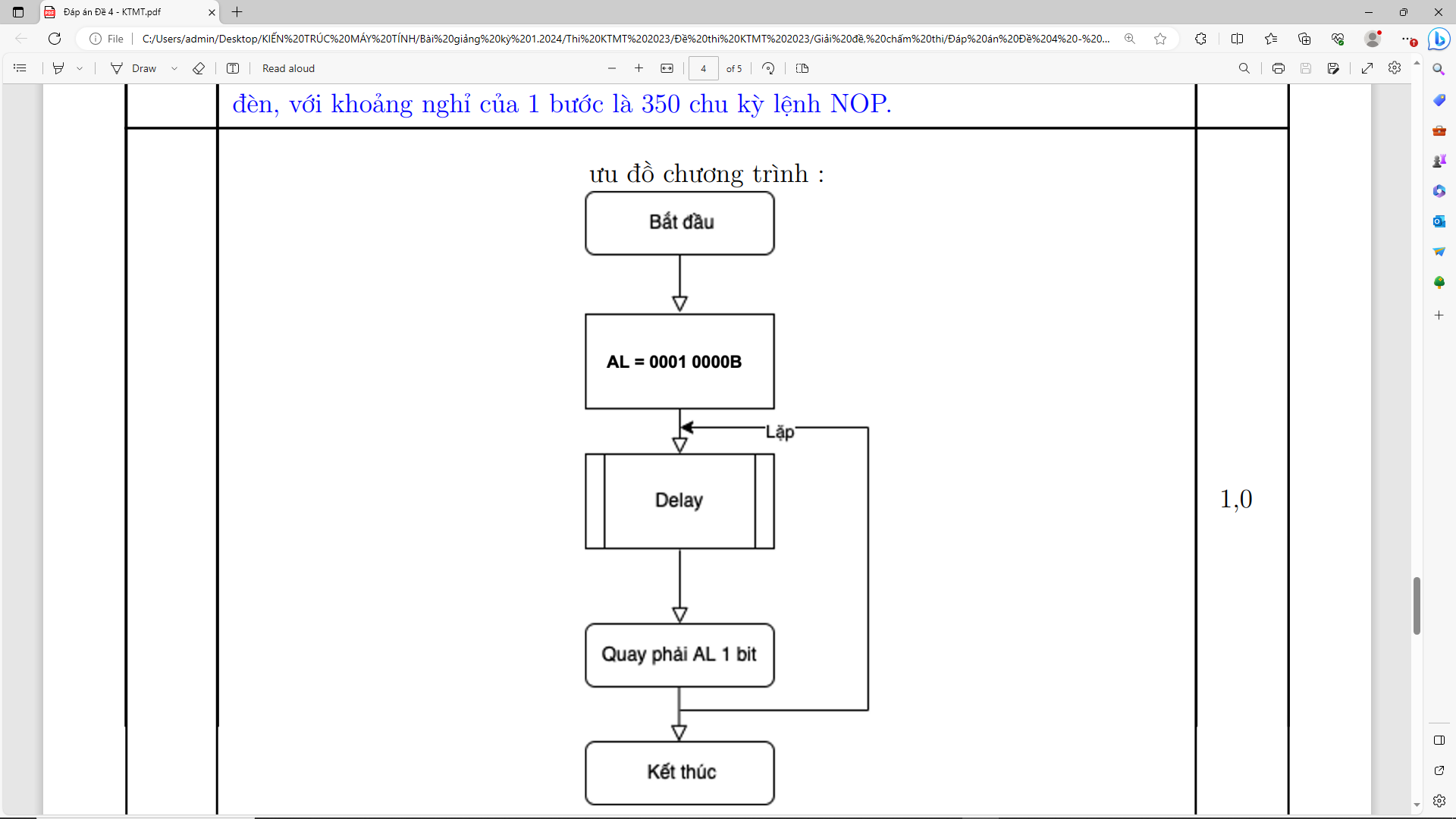
INT 21H

MAIN Endp

END MAIN

**8. Mạch điều khiển 8 đèn LED (D0-D7) nối với hệ vi xử lý 8086 tại cổng ra 0D21H. Biết rằng đèn được bật sáng nếu bít điều khiển là 1. Ngược lại bằng 0 thì đèn sẽ tắt. Vẽ lưu đồ và viết chương trình hợp ngữ tạo hiệu ứng : một đèn sáng chạy từng bước từ trái sang phải, bắt đầu từ đèn D3, mỗi bước dịch chuyển tương ứng với 1 đèn, với khoảng nghỉ của 1 bước là 350 chu kỳ lệnh NOP.**

*Lưu đồ chương trình:*



**Chương trình hợp ngữ:**

.Model small

.Stack 100H

.Data

DK\_LED EQU 0D21H ; Cong dieu khien den LED

.code

Main Proc

MOV AX, @Data ; khoi tao dau thanh ghi DS

MOV DS, AX

MOV AL, 10H ; Bat dau tu den D3: 0001 0000B

MOV DX, DK\_LED

LAP:

OUT DX, AL

MOV CX, 350 ;Tre 64 chu ky NOP

TRE\_350: NOP

LOOP TRE\_350

ROR AL, 1 ; Nhay dich vong sang phai 1 bit

JMP LAP

MOV AH, 4CH

INT 21H

MAIN Endp

END MAIN

**9. Vẽ lưu đồ thuật toán và Viết chương trình hợp ngữ để nhập nhiệt độ cận dưới của 1 hệ thống điều khiển nhiệt độ có giá trị nằm giữa 30 và 40. Trên lưu đồ thuật toán cần đánh dấu các nhãn của chương trình hợp ngữ.**

.Model small

Bắt đầu

Đọc t0

**t0<300**

**t0>400**

Tắt đèn

Bật đèn

Đ

S

Trễ 1000NOP

.Stack 100h

.Data

DK\_Nhietdo EQU 0D21H

.code

Main Proc

MOV AX,@DATA

MOV DS,AX

MOV AL,200

MOV DX,DK\_Nhietdo

CMP AL,30

JB NH30

CMP AL,40

JA LH40

JMP Xong1lan

NH30:

MOV AL,1

OUT 100,AL

JMP Xong1lan

LH40:

MOV AL,1

OUT 100,AL

JMP Xong1lan

Xong1lan:

MOV CX,100

Delay:NOP

LOOP Delay

MOV AH,4CH

INT 21H

MAIN Endp

END MAIN

**III. GHÉP NỐI CPU VỚI BỘ NHỚ VÀ THIẾT BỊ NGOẠI VI**

**1.** **Xây dựng bộ giải mã địa chỉ cho một bộ nhớ ROM có dung lượng 64KB có địa chỉ bắt đầu là 2C000H với các chíp nhớ có dung lượng 16Kx8. Chỉ được sử dụng các chip giải mã địa chỉ 74LS138 (là các chip giải mã có 3 đầu vào và 8 đầu ra)**

***B1. Xác định số bit cho địa chỉ nội bộ chip và mạch giải mã:***

*+ Chíp nhớ IC 16K×8 chiếm không gian 16KB = 214B ⇒ 14 bit cho địa chỉ nội*

*bộ chíp (A0 − A13)*

*+ Cần 20 - 14 = 6 bit cho mạch giải mã (A14 − A19)*

***B2. Phân giải địa chỉ cơ sở của các chip:***

*+ Số lượng chip nhớ 16K×8 là 64KB/16KB = 4*

*+ Dung lượng một chíp nhớ là 04000H*

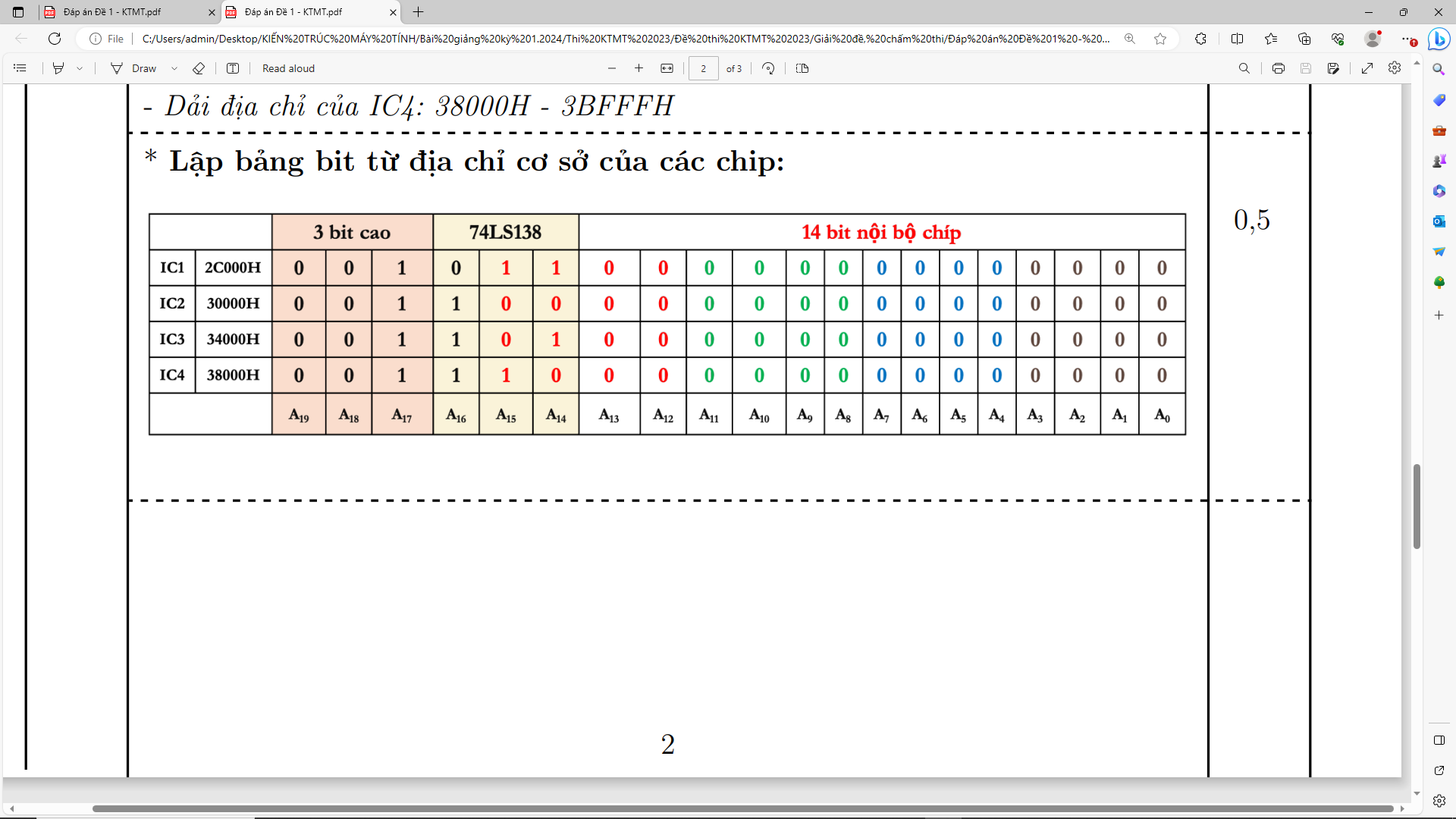
*- Dải địa chỉ của IC1: 2C000H - 2FFFFH*

*- Dải địa chỉ của IC2: 30000H - 33FFFH*

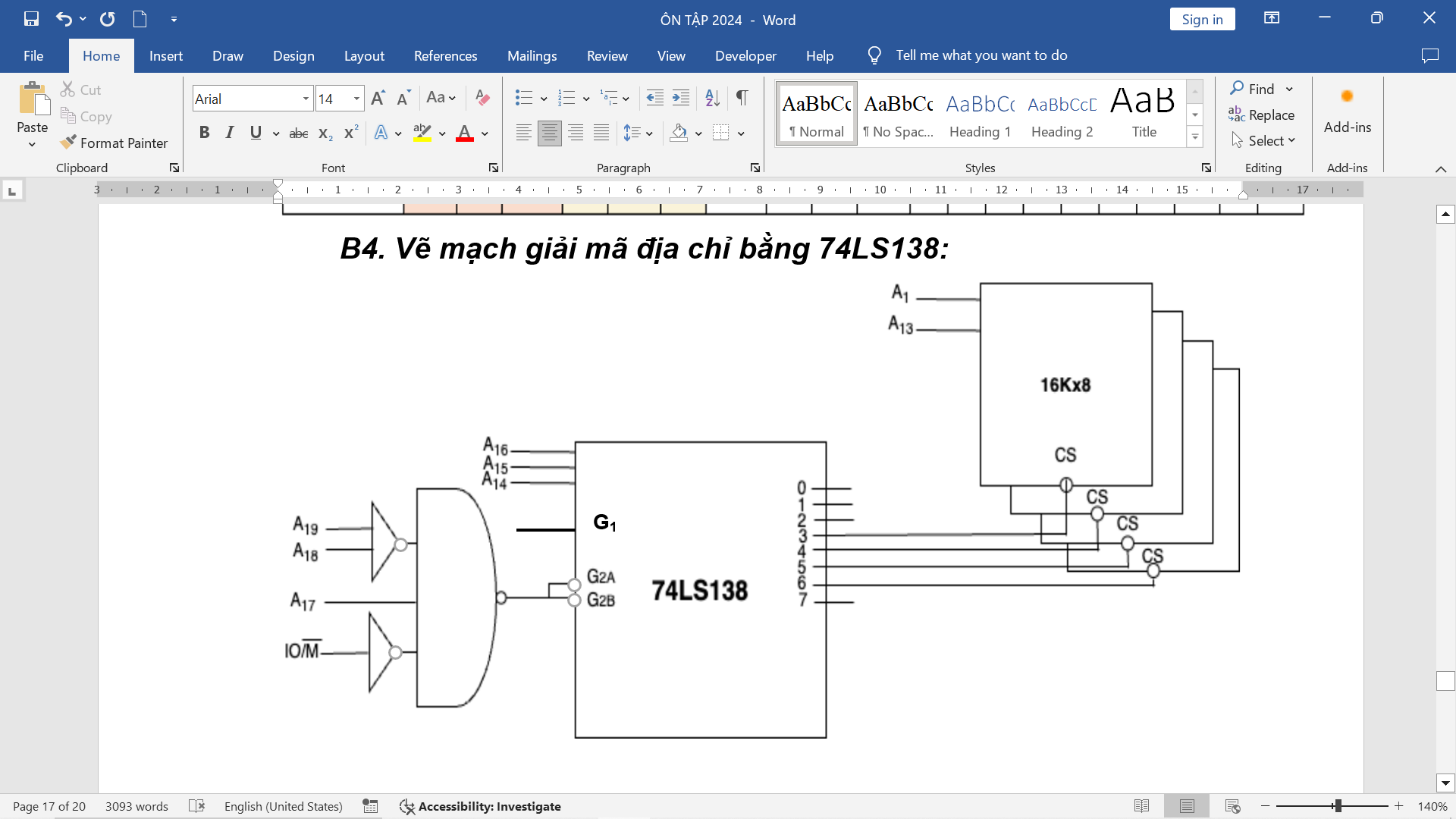
*- Dải địa chỉ của IC3: 34000H - 37FFFH*

*- Dải địa chỉ của IC4: 38000H - 3BFFFH*

***B3. Lập bảng bit từ địa chỉ cơ sở của các chip:***



***B4. Vẽ mạch giải mã địa chỉ bằng 74LS138:***



**2. Hãy xây dựng mạch giải mã địa chỉ cho một bộ nhớ ROM có dung lượng 128 KB bằng phương pháp sử dụng các cổng logic; biết rằng kích thước vi mạch nhớ là 32KB và địa chỉ cơ sở là 58000H.**

***B1. Xác định số bit cho địa chỉ nội bộ chip và mạch giải mã:***

+ Chíp nhớ IC 32Kx8 chiếm không gian 32KB = 215B

⇒ 15 bit cho địa chỉ nội bộ chíp (A0 − A14)

+ Cần 20 - 15 = 5 bit cho mạch giải mã (A15 − A19)

***B2. Phân giải địa chỉ cơ sở của các chip:***

+ Số lượng chip nhớ 32K×8 là 128KB/32KB = 4

+ Dung lượng một chíp nhớ là 08000H

Dải địa chỉ

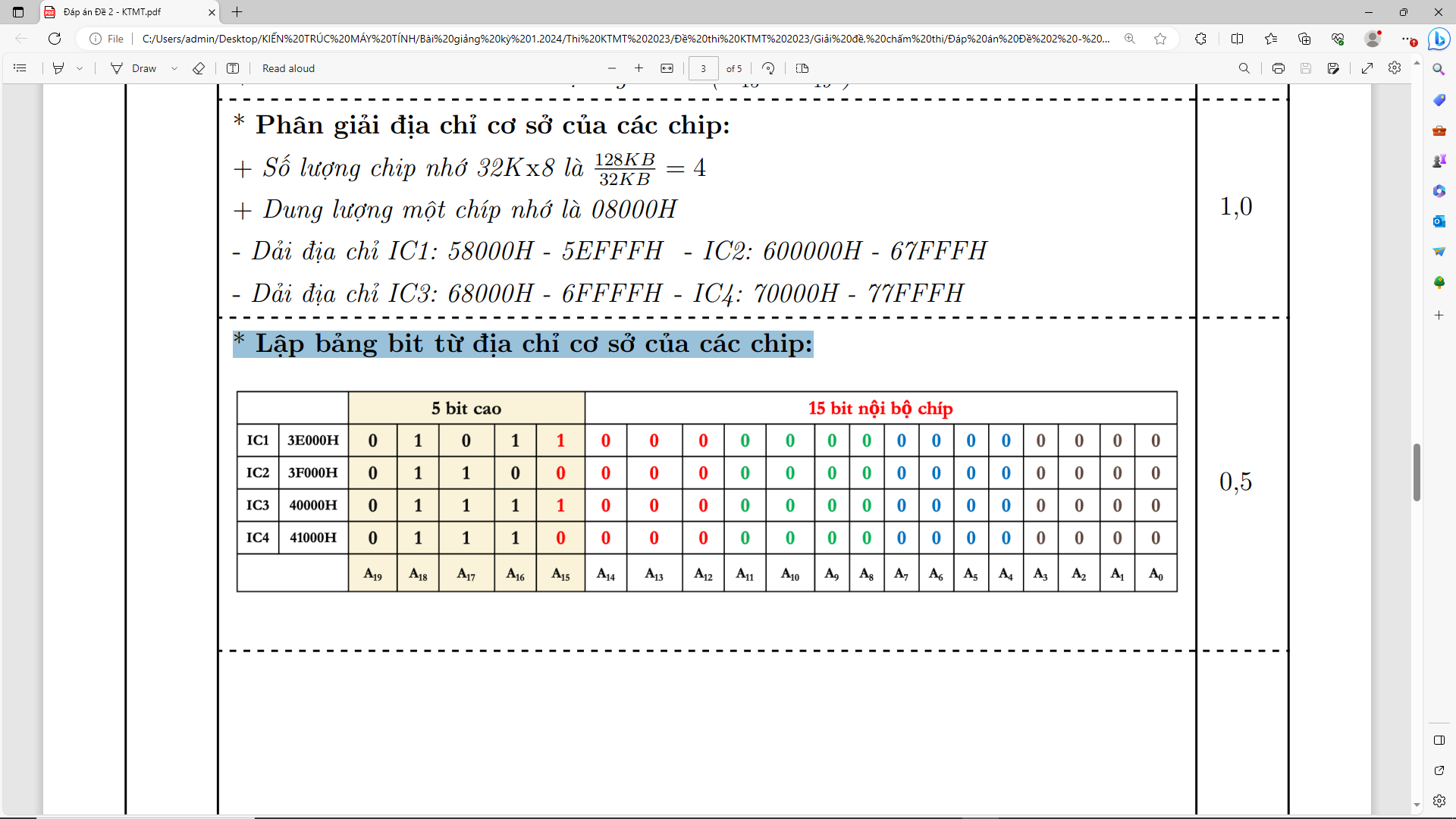
- IC1: 58000H - 5EFFFH

- IC2: 600000H - 67FFFH

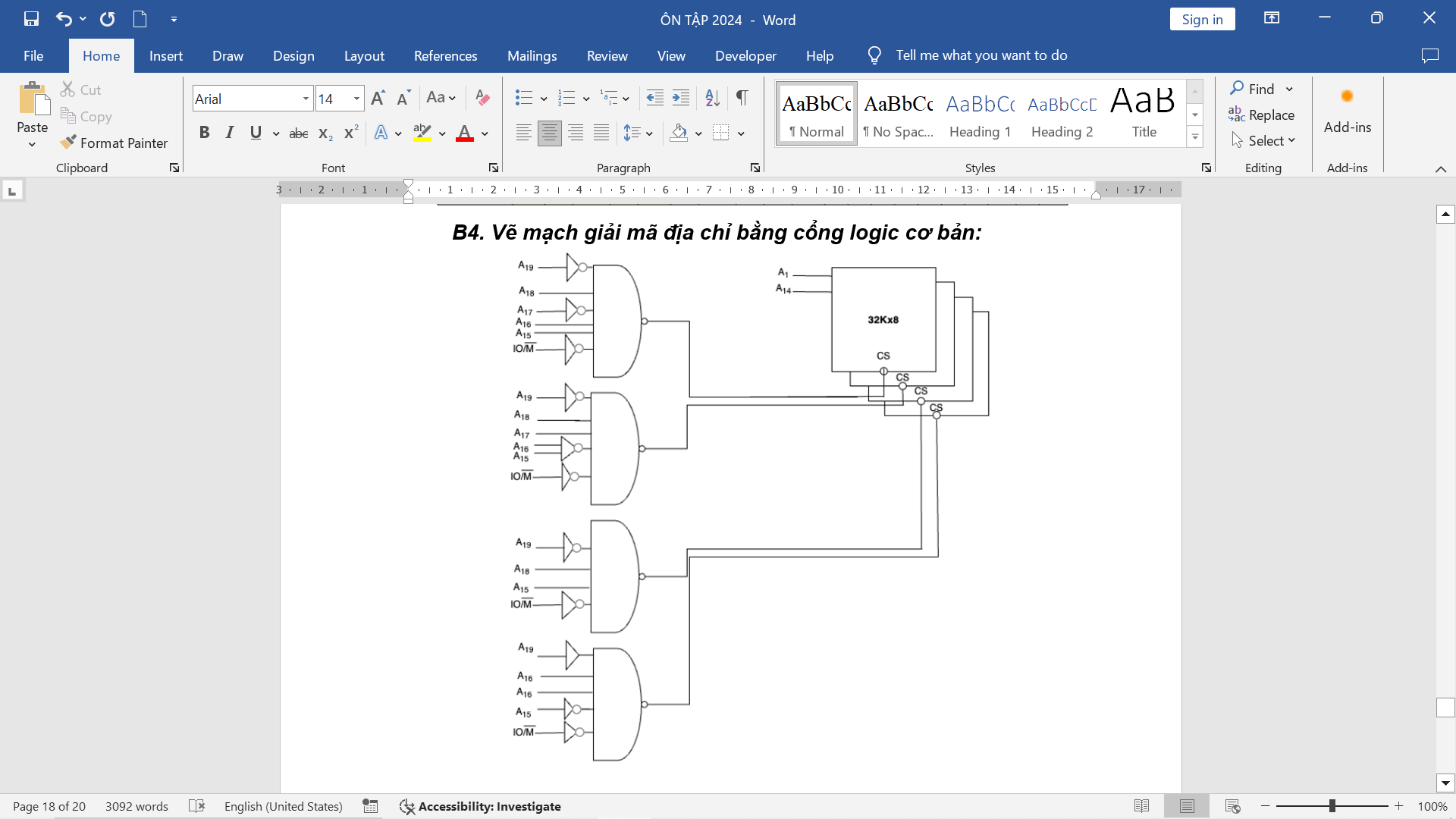
- Dải địa chỉ IC3: 68000H - 6FFFFH

- IC4: 70000H - 77FFFH

***B3. Lập bảng bit từ địa chỉ cơ sở của các chip:***



***B4. Vẽ mạch giải mã địa chỉ bằng cổng logic cơ bản:***



**3. Xây dựng bộ giải mã địa chỉ bộ nhớ có dung lượng 16KB có địa chỉ bắt đầu là 3E000H với các chíp nhớ có dung lượng 4Kx8. Chỉ được sử dụng các chip giải mã địa chỉ 74LS139 (là các chip giải mã có 2 đầu vào và 4 đầu ra).**

***B1. Xác định số bit cho địa chỉ nội bộ chip và mạch giải mã:***

+ Chíp nhớ IC 4K×8 chiếm không gian 4KB = 212B =⇒ 12 bit cho địa chỉ nội bộ chíp (A0 − A11)

+ Cần 20 - 12 = 8 bit cho mạch giải mã (A12 − A19)

***B2. Phân giải địa chỉ cơ sở của các chip:***

+ Số lượng chip nhớ 4K×8 là 16KB/4KB = 4

+ Dung lượng một chíp nhớ là 01000H

- Dải địa chỉ

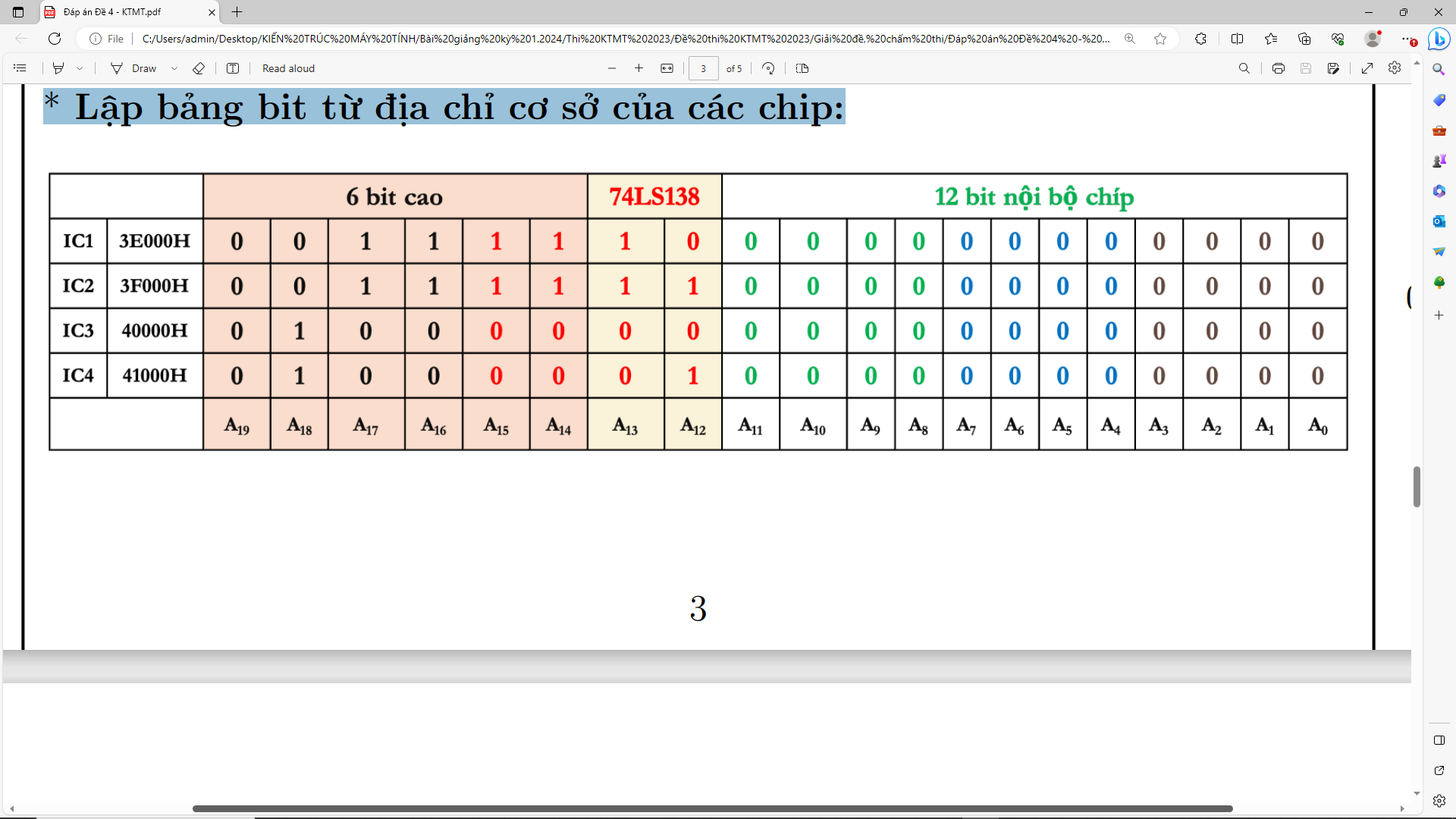
IC1: 3E000H - 3EFFFH

IC2: 3F0000H - 3FFFFH

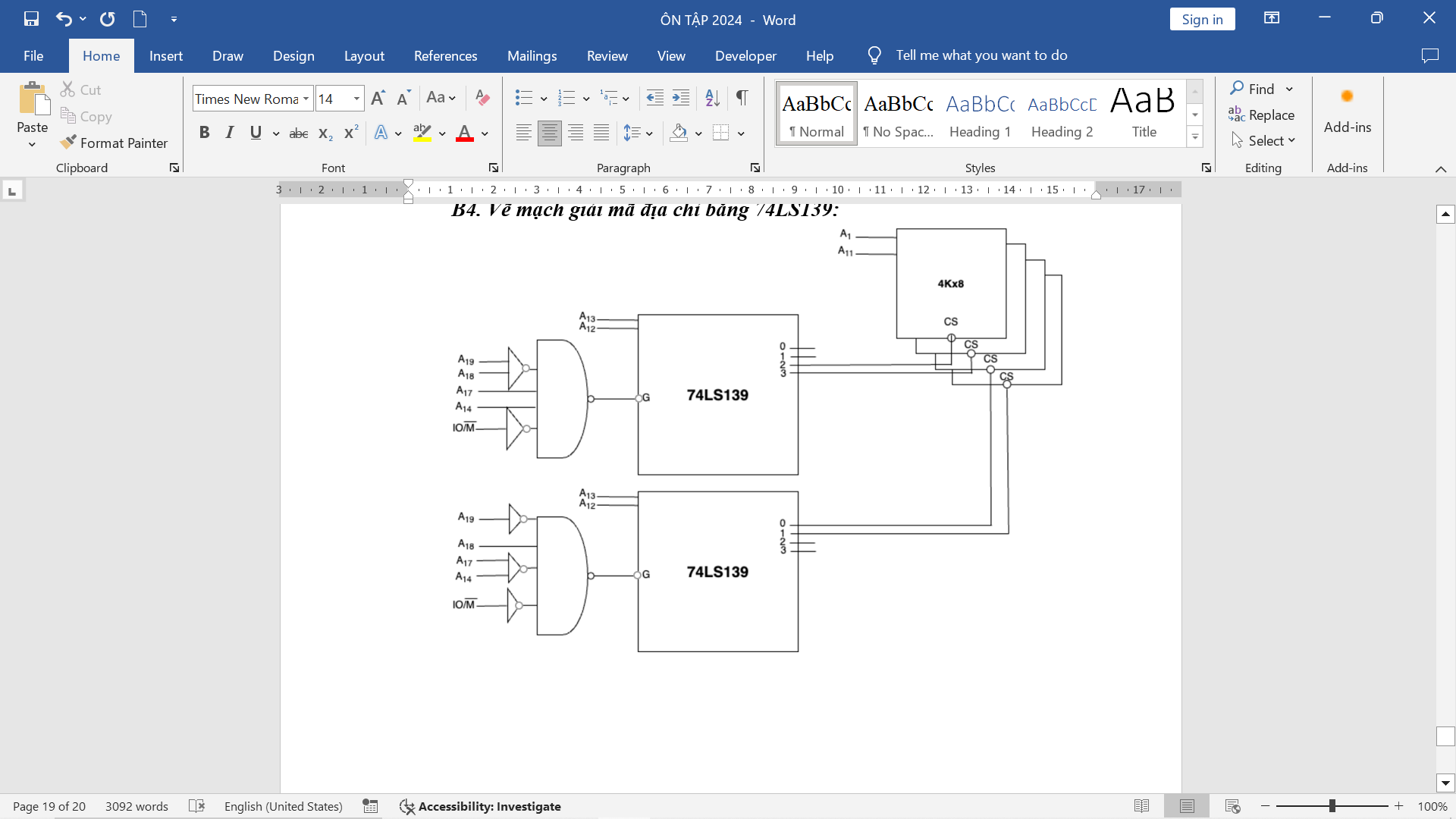
IC3: 40000H - 40FFFH

IC4: 41000H - 41FFFH

***B3. Lập bảng bit từ địa chỉ cơ sở của các chip:***



***B4. Vẽ mạch giải mã địa chỉ bằng 74LS139:***



**IV. XEM THÊM**

**LÝ THUYẾT**

1. Trình bày khái niệm Kiến trúc máy tính và các thành phần của Kiến trúc máy tính.

2. Trình bày các thanh ghi của vi xử lý Intel 8086.

3. Trình bày các thành phần cơ bản của tổ chức máy tính.

4. Trình bày khuôn dạng lệnh của vi xử lý Intel 8086

5. Trình bày cấu trúc của bộ xử lý trung tâm.

6. Trình bày hệ thống bộ nhớ phân cấp trong các hệ thống máy tính

7. Phân biệt hai loại máy tính CISC và RISC .

8. Trình bày 2 nguyên lý hoạt động tạo ra cơ chế hoạt động thông minh của bộ nhớ cache.

9. Cơ chế ống lệnh (pipeline) của CPU thường gặp phải những vấn đề gì? Nêu một hướng giải quyết xung đột dữ liệu trong pipeline khi thực hiện đoạn chương trình sau:

ADD R1, R2, R3 ; R1 <== R2+R3

ADD R4, R4, #300 ; R4 <== R4+300

CMP R1, #100 ; so sánh R1 với 100

SUB R5, #2000 ; R5 <== R5 + 2000

biết rằng mỗi lệnh được chia thành 5 giai đoạn trong pipeline: Đọc lệnh (IF), giải mã & đọc toán hạng (ID), truy nhập bộ nhớ (MEM), thực hiện (EX) và lưu kết quả (WB).

10. Cơ chế ống lệnh (pipeline) của CPU thường gặp phải những vấn đề gì? Nêu một hướng giải quyết xung đột dữ liệu trong pipeline khi thực hiện đoạn chương trình sau:

ADD R4, R4, #300 ; R4 <== R4+300

ADD R1, R1, R3 ; R1 <== R1+R3

SUB R5, #2000 ; R5 <== R5 + 2000

SUB R1, R1, #100 ; R1 <== R1 - 100

biết rằng mỗi lệnh được chia thành 5 giai đoạn trong pipeline: Đọc lệnh (IF), giải mã & đọc toán hạng (ID), truy nhập bộ nhớ (MEM), thực hiện (EX) và lưu kết quả (WB).

11. Cho đoạn chương trình sau (R1, R2 là các thanh ghi và các lệnh quy ước theo dạng LỆNH <ĐÍCH> <GỐC>):

(1) LOAD R2, #400

(2) LOAD R1, #1200

(3) STORE (R1), R2

(4) SUBSTRACT R2, #20

(5) ADD 1200, #10

(6) ADD R2, (R1)

a. Xác định chế độ địa chỉ và ý nghĩa của từng lệnh;

b. Xác định giá trị của thanh ghi R2 sau khi thực hiện xong lệnh số (6).

12. Cho đoạn chương trình sau (R1, R2 là các thanh ghi và các lệnh quy ước theo dạng LỆNH <ĐÍCH> <GỐC>):

(1) LOAD R2, #500

(2) LOAD R1, #2000

(3) STORE (R1), R2

(4) ADD 2000, #30

(5) SUBSTRACT R2, #15

(6) ADD R2, (R1)

a. Xác định chế độ địa chỉ và ý nghĩa của từng lệnh;

b. Xác định giá trị của thanh ghi R2 sau khi thực hiện xong lệnh số (6).

**BÀI TẬP**

1. Cho một máy tính có độ rộng bus dữ liệu là 32 bit, quản lý được bộ nhớ có dung lượng tối đa là 64GB, bộ nhớ cache có dung lượng 4MB với dòng cache 64KB. Hãy xác định các thành phần địa chỉ của bộ nhớ khi sử dụng phương pháp ánh xạ trực tiếp.

2. Cho một máy tính có độ rộng bus dữ liệu là 64 bit, quản lý được bộ nhớ có dung lượng tối đa là 128GB, bộ nhớ cache có dung lượng 4MB và 4 đường cache với dòng cache 32KB. Hãy xác định các thành phần địa chỉ của bộ nhớ khi sử dụng phương pháp ánh xạ tập kết hợp.

3. Cho một máy tính có độ rộng bus dữ liệu là 32 bit, quản lý được bộ nhớ có dung lượng tối đa là 64GB, bộ nhớ cache có dung lượng 4MB với dòng cache 64KB. Hãy xác định các thành phần địa chỉ của bộ nhớ khi sử dụng phương pháp ánh xạ trực tiếp.

4. Thiết kế bộ giải mã địa chỉ cho bộ nhớ có dung lượng 16KB từ các IC nhớ có dung lượng 4KB. Bộ nhớ có địa chỉ cơ sở là 79000H. Bộ giải mã địa chỉ được chế tạo bằng các mạch logic tổ hợp.

5. Thiết kế bộ giải mã địa chỉ cho bộ nhớ có dung lượng 16KB từ các IC nhớ có dung lượng 4KB. Bộ nhớ có địa chỉ cơ sở là 97000H. Bộ giải mã địa chỉ được chế tạo bằng các cổng logic cơ bản.

6. Thiết kế bộ giải mã địa chỉ cho bộ nhớ có dung lượng 16KB từ các IC nhớ có dung lượng 4KB. Bộ nhớ có địa chỉ cơ sở là 79000H. Bộ giải mã địa chỉ được chế tạo bằng các mạch logic tổ hợp.

7. Vẽ lưu đồ thuật toán và Viết chương trình hợp ngữ nhập vào 1 chuỗi ký tự không quá 100 ký tự, chuyển đổi các ký tự chữ thường thành chữ in, lưu vào bộ nhớ và in chuỗi ký tự ra màn hình. Trên lưu đồ thuật toán cần đánh dấu các nhãn của chương trình hợp ngữ.

8. Vẽ lưu đồ thuật toán và Viết chương trình hợp ngữ để nhập nhiệt độ cận dưới của 1 hệ thống điều khiển nhiệt độ có giá trị nằm giữa 30 và 40. Trên lưu đồ thuật toán cần đánh dấu các nhãn của chương trình hợp ngữ.

9. Vẽ lưu đồ thuật toán và Viết chương trình hợp ngữ chuyển đổi các ký tự từ chữ thường sang chữ in trong bộ nhớ có địa chỉ [1234H:2356H] và độ dài của chuỗi ký tự là 100, không có ký tự $ trong bộ nhớ. Sau khi chuyển đổi in chuỗi ký tự ra màn hình. Trên lưu đồ thuật toán cần đánh dấu các nhãn của chương trình hợp ngữ.