|  |
| --- |
| **BỘ THÔNG TIN VÀ TRUYỀN THÔNG** |
| **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  **-------------------------------------** |
|  |
|  |
| **ĐỒ ÁN MÔN HỌC**  **KỸ THUẬT VI XỬ LÝ** |
|  |
| ***Đề tài***: “*Mạch Cảnh Báo Cháy Tự Động”* |
|  |
|  |
|  |
|  |
|  |
| **Giảng viên hướng dẫn : Nguyễn Quý Sỹ** |
| **Sinh viên thực hiện :**  **Nguyễn Quang Liêm\_MSV:B18DCCN325** |
| **Hoàng Khắc Đoàn\_MSV:B18DCCN156**  **Vũ Tiến Phúc\_MSV:B18DCCN475** |
|  |
|  |
|  |
| **HÀ NỘI – NĂM 2020** |

NỘI DUNG ĐỒ ÁN GỒM

I. Mục tiêu.

II. Trình bày vi xử lý 8086

1.Cấu trúc 8086

2. Thành phần chức năng

3.Các ngắt

4.Tập lệnh

5. Các chế độ địa chỉ

III. Trình bày thiết kế hệ thống

1.Cấu trúc hệ thống

2 .Sơ đồ khối hệ thống

3.Cấu tạo và nguyên lý hoạt động của các linh kiện

4 Nguyên lí hoạt động của hệ thống và sơ đồ nguyên lý

IV. Thuật toán điều khiển

1.Lưu đồ giải thuật

2.Giải thích lưu đồ.

**I. MỤC TIÊU**

Hiện nay các vấn đề cháy nổ ở các nhà hàng, khu chung cư, khu công nghiệp,

nhà kho đang rất được quan tâm. Chính về thế chúng em nghiên cứu về hệ thống

báo cháy nhằm phục vụ cho vấn đề này.

Hệ thống tự động cảnh báo cháy nhằm cung cấp cho người sử dụng biết nhiệt độ

hiện tại và cảnh báo khi nhiệt độ quá cao ngây nguy hiển cho các nhà máy phân

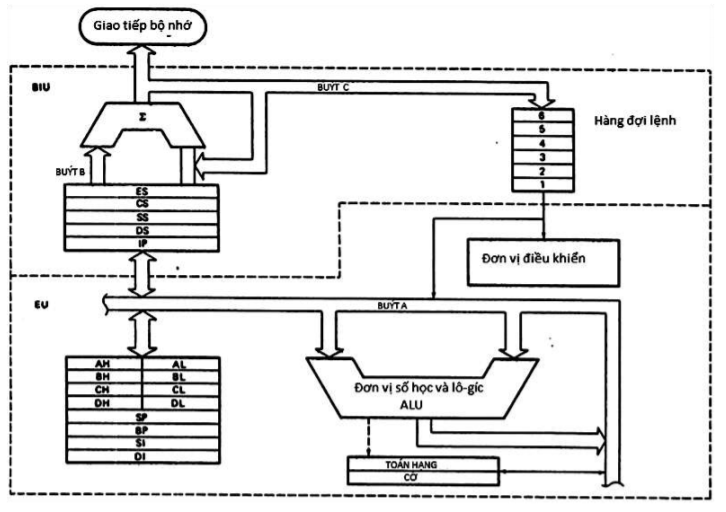
xưởng các kho chứ dễ cháy nổ khi nhiệt độ cao….tránh gây thiệt hại cho người và của.

**II.TRÌNH BÀY VI XỬ LÝ 8086**

Intel 8086 là bộ vi xử lý 16 bít đầu tiên của Intel và là vi xử lý đầu tiên hỗ trợ tập lệnh x86. Vi xử lý được sử dụng trong nhiều lĩnh vực khác nhau, nhất là trong các máy IBM PC/XT. Các bộ vi xử lý thuộc họ này sẽ còn được sử dụng rộng rãi trong thời gian tới do tính kế thừa của các sản phẩm trong họ x86. Các chương trình viết cho 8086 vẫn có thể chạy trên các hệ thống tiên tiến sau này.

**1.Cấu trúc**

Sơ đồ khối cấu trúc của vi xử lý 8086:



**2.Thành phần, chức năng**

1. **Đơn vị giao tiếp buýt và thực thi EU:**

EU bao gồm một đơn vị điều khiển, khối này có mạch giải mã lệnh. Mã lệnh đọc vào từ bộ nhớ được đưa đến đầu vào của bộ giải mã, các thông tin thu được từ đầu ra của nó sẽ được đưa đến mạch tạo xung điều khiển, kết quả là ta thu được các dãy xung khác nhau trên kênh điều khiển (tuỳ theo mã lệnh) để điều khiển hoạt động của các bộ phận bên trong và bên ngoài CPU. Ngoài ra, EU còn có khối số học và lôgic (Arithmetic and Logic Unit - ALU) dùng để thực hiện các thao tác khác nhau với các toán hạng của lệnh. Tóm lại, khi CPU hoạt động EU sẽ cung cấp thông tin về địa chỉ cho BIU để khối này đọc lệnh và dữ liệu, còn bản thân nó thì đọc lệnh và giải mã lệnh.

1. **Các thanh ghi**:

**\*Thanh ghi đoạn:** Bộ vi xử lý 8086 cung cấp các các thanh ghi 16 bít liên quan đến địa chỉ đầu của các đoạn, chúng được gọi là các thanh ghi đoạn (Segment Registers) cụ thể:

− Thanh ghi đoạn mã CS (Code-Segment)

− Thanh ghi đoạn dữ liệu DS (Data Segment)

− Thanh ghi đoạn ngăn xếp SS (Stack Segment)

− Thanh ghi đoạn dữ liệu phụ ES (Extra Segment).

Các thanh ghi đoạn 16 bít này chỉ ra địa chỉ đầu của bốn đoạn trong bộ nhớ, dung lượng lớn nhất của mỗi đoạn nhớ này là 64 KByte và tại một thời điểm nhất định bộ vi xử lý chỉ làm việc được với bốn đoạn nhớ 64 KByte này. Để xác định chính xác vị trí một ô nhớ của chương trình các thanh ghi đoạn sẽ phải phối hợp với các thanh ghi đặc biệt khác còn gọi là các thanh ghi lệch hay phân đoạn (offset register).

**\* Các thanh ghi đa năng:** Trong khối EU có bốn thanh ghi đa năng 16 bít AX, BX, CX, DX. Điều đặc biệt là khi cần chứa các dữ liệu 8 bít thì mỗi thanh ghi có thể tách ra thành hai thanh ghi 8 bít cao và thấp để làm việc độc lập, đó là các tập thanh ghi AH và AL, BH và BL, CH và CL, DH và DL (trong đó H chỉ phần cao, L chỉ phần thấp). Mỗi thanh ghi có thể dùng một cách vạn năng để chứa các tập dữ liệu khác nhau nhưng cũng có công việc đặc biệt nhất định chỉ thao tác với một vài thanh ghi nào đó. Chính vì vậy các thanh ghi thường được gán cho những cái tên có ý nghĩa. Cụ thể:

− AX (accumulator): thanh ghi tích lũy. Các kết quả của các thao tác thường được chứa ở đây (kết quả của phép nhân, chia). Nếu kết quả là 8 bít thì thanh ghi AL được coi là thanh ghi chứa.

− BX (base): thanh ghi cơ sở thường chứa địa chỉ cơ sở của một bảng.

− CX (count): bộ đếm. CX thường được dùng để chứa số lần lặp trong trường hợp các lệnh LOOP (lặp), còn CL thường cho ta số lần dịch hoặc quay trong các lệnh dịch hoặc quay thanh ghi.

− DX (data): thanh ghi dữ liệu DX cùng BX tham gia các thao tác của phép nhân hoặc chia các số 16 bít. DX thường dùng để chứa địa chỉ của các cổng trong các lệnh vào/ ra dữ liệu trực tiếp.

**\*** **Các thanh ghi con trỏ và chỉ số**: Trong 8086 còn có ba thanh ghi con trỏ và hai thanh ghi chỉ số 16 bít. Các thanh ghi này (trừ IP) đều có thể được dùng như các thanh ghi đa năng, nhưng ứng dụng chính của mỗi thanh ghi là chúng được ngầm định như là thanh ghi lệch cho các đoạn tương ứng. Cụ thể:

− IP: con trỏ lệnh (Instruction Pointer). IP luôn trỏ vào lệnh tiếp theo sẽ được thực hiện nằm trong đoạn mã CS. Địa chỉ đầy đủ của lệnh tiếp theo này ứng với CS:IP.

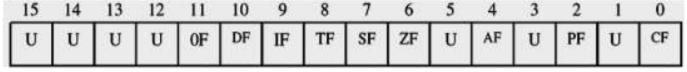
− BP: con trỏ cơ sở (Base Pointer). BP luôn trỏ vào một dữ liệu nằm trong đoạn ngăn xếp SS. Địa chỉ đầy đủ của một phần tử trong đoạn ngăn xếp ứng với SS:BP.

− SP: con trỏ ngăn xếp (Stack Pointer). SP luôn trỏ vào đỉnh hiện thời của ngăn xếp nằm trong đoạn ngăn xếp SS. Địa chỉ đỉnh ngăn xếp ứng với SS:SP.

− SI: chỉ số gốc hay nguồn (Source Index). SI chỉ vào dữ liệu trong đoạn dữ liệu DS mà địa chỉ cụ thể đầy đủ ứng với DS:SI.

− DI: chỉ số đích (Destination Index). DI chỉ vào dữ liệu trong đoạn dữ liệu DS mà địa chỉ cụ thể đầy đủ ứng với DS:DI.

**\* Thanh ghi cờ FR (flag register):** Đây là thanh ghi khá đặc biệt trong CPU, mỗi bít của nó được dùng để phản ánh một trạng thái nhất định của kết quả phép toán do ALU thực hiện hoặc một trạng thái hoạt động của EU. Dựa vào các cờ này người lập trình có thể có các lệnh thích hợp tiếp theo cho bộ vi xử lý (các lệnh nhảy có điều kiện). Thanh ghi cờ gồm 16 bít nhưng người ta chỉ dùng hết 9 bít của nó để làm các bít cờ:



− U không sử dụng.

− C hoặc CF (Carry Flag): cờ nhớ. CF = 1 khi có nhớ hoặc muợn từ bít có nghĩa lớn nhất MSB (Most Significant Bit).

− P hoặc PF (Parity Flag): cờ parity. PF phản ánh tính chẵn lẻ của tổng số bít 1 có trong kết quả. Cờ PF =1 khi tổng số bít 1 trong kết quả là chẵn (even parity).

− A hoặc AF (Auxiliary Carry Flag): cờ nhớ phụ rất có ý nghĩa khi ta làm việc với các số BCD (Binary Coded Decimal). AF = 1 khi có nhớ hoặc muợn từ một số BCD thấp (4 bít thấp) sang một số BCD cao (4 bít cao).

− Z hoặc ZF (Zero Flag): cờ rỗng. ZF =1 khi kết quả = 0.

− S hoặc SF (sign flag): cờ dấu. SF = 1 khi kết quả âm.

− O hoặc OF (Overflow Flag): cò tràn. OF = 1 khi kết quả là một số bù 2 vượt qua ngoài giới hạn biểu diễn dành cho nó.

Đây là 6 bít cờ trạng thái phản ánh các trạng thái khác nhau của kết sau một thao tác nào đó, trong đó 5 bít cờ đầu thuộc byte thấp của thanh cờ là các cờ giống như của bộ vi xử lý 8 bít 8085 của Intel. Chúng được lập hoặc xoá tuỳ theo các điều kiện cụ thể sau các thao tác của ALU.

Ngoài ra, bộ vi xử lý 8086 còn có các cờ điều khiển sau đây (các cờ này được lập hoặc xoá bằng các lệnh riêng):

− T hoặc TF (Trap Flag): cờ bẫy. TF = 1 thì CPU làm việc ở chế độ chạy từng lệnh (chế độ này dùng khi cần tìm lỗi trong một chương trình).

− I hoặc IF (Interrupt Enable Flag): cờ cho phép ngắt. IF = 1 thì CPU cho phép các yêu cầu ngắt (che được) được tác động.

− D hoặc DF (Direction Flag): cờ hướng. DF = 1 khi CPU làm việc với chuỗi ký tự theo thứ tự từ phải sang trái (vì vậy D chính là cờ lùi).

**3.Các ngắt**

* Ngắt là việc tạm dừng việc chương trình đang chạy để CPU có thể chạy một chương trình khác nhằm xử lý một yêu cầu do bên ngoài đưa tới CPU như yêu cầu vào/ra hoặc do chính yêu cầu của bên trong CPU như lỗi trong khi tính toán.
* Trong hệ vi xử lý 8086 có thể xếp các nguyên nhân gây ra ngắt CPU vào 3 nhóm như sau:

+ Nhóm các ngắt cứng: đó là các yêu cầu ngắt CPU do các tín hiệu đến từ các chân INTR và NMI. Ngắt cứng INTR là yêu cầu ngắt che được. Các lệnh CLI và STI có ảnh hưởng trực tiếp tới trạng thái của cờ IF trong bộ vi xử lý, tức là ảnh hưởng tới việc CPU có nhận biết yêu cầu ngắt tại chân này hay không. Yêu cầu ngắt tại chân INTR có thể có kiểu ngắt N nằm trong khoảng 0-FFH. Kiểu ngắt này phải được đưa vào buýt dữ liệu để CPU có thể đọc được khi có xung trong chu kỳ trả lời chấp nhận ngắt.

+ Nhóm các ngắt mềm: khi CPU thực hiện các lệnh ngắt dạng INT N, trong đó N là số hiệu (kiểu) ngắt nằm trong khoảng 00-FFH (0-255).

+ Nhóm các hiện tượng ngoại lệ: đó là các ngắt do các lỗi nảy sinh trong quá trình hoạt động của CPU như phép chia cho 0, xảy ra tràn khi tính toán.

* Yêu cầu ngắt sẽ được CPU kiểm tra thường xuyên tại chu kỳ đồng hồ cuối cùng của mỗi lệnh. Error! Reference source not found. trình bày một cách đơn giản để đưa được số hiệu ngắt N vào buýt dữ liệu trong khi cũng tạo ra yêu cầu ngắt đưa vào chân INTR của bộ vi xử lý 8086. Giả thiết trong một thời điểm nhất định chỉ có một yêu cầu ngắt IRi được tác động và sẽ có xung yêu cầu ngắt đến CPU. Tín hiệu IRi được đồng
* Đáp ứng của CPU khi có yêu cầu ngắt :

+ Khi có yêu cầu ngắt kiểu N đến CPU và nếu yêu cầu đó được phép, CPU thực hiện các công việc sau:

1*.* SP ← SP-2, [SP] ← FR, trong đó [SP] là ô nhớ do SP chỉ ra. (chỉ ra đỉnh mới của ngắn xếp, cất thanh ghi cờ vào đỉnh ngăn xếp)

2. IF ← 0, TF ← 0. (cấm các ngắt khác tác động vào CPU, cho CPU chạy ở chế độ bình thường)

3*.* SP ← SP-2, [SP] ← CS. (chỉ ra đỉnh mới của ngăn xếp, cất phần địa chỉ đoạn của địa chỉ trở về vào đỉnh ngăn xếp)

4*.* SP ← SP-2, [SP] ← IP (chỉ ra đỉnh mới của ngăn xếp, cất phần địa chỉ lệch của địa chỉ trờ về vào đỉnh ngăn xếp)

5*.* [N\*4] → IP, [N\*4+2] → CS (lấy lệnh tại địa chỉ mới của chương trình con phục vụ ngắt kiểu N tương ứng trong bảng vectơ ngắt)

6. Tại cuối chương trình phục vụ ngắt, khi gặp lệnh IRET [SP] → IP, SP ← SP+2 [SP] → CS, SP ← SP+2 [SP] → FR, SP ← SP+2 (bộ vi xử lý quay lại chương trình chính tại địa chỉ trở về và với giá trị cũ của thanh ghi cờ được lấy ra từ ngăn xếp).

* Một số kiểu ngắt đặc biệt được xếp vào đầu dãy ngắt mềm INT N:

|  |  |
| --- | --- |
| INT 0 | Ngắt mềm do phép chia cho số 0 gây ra |
| INT 1 | Ngắt mềm để chạy từng lệnh ứng với trường hợp cờ TF=1 |
| INT 2 | Ngắt cứng do tín hiệu tích cực tại chân NMI gây ra |
| INT 3 | Ngắt mềm để đặt điểm dừng của chương trình tại một địa chỉ nào đó |
| INT 4 | Ngắt mềm ứng với trường hợp cờ tràn OF=1. |
| INT 5-INT 1FH | Dành riêng cho Intel trong các bộ vi xử lý cao cấp khác. |
| INT 20H-INT FFH | Dành cho người sử dụng. |

* Các kiểu ngắt N trong INT N đều tương ứng với các địa chỉ xác định của chương trình con phục vụ ngắt mà ta có thể tra được trong bảng các vectơ ngắt. Intel quy định bảng này nằm trong RAM bắt đầu từ địa chỉ 00000H và dài 1 KB (vi xử lý 8086 có tất cả kiểu ngắt, mỗi kiểu ngắt ứng với 1 vectơ ngắt, 1 vectơ ngắt cần 4 byte để chứa địa chỉ đầy đủ cho CS:IP của chương trình con phục vụ ngắt).
* Xử lý ưu tiên khi ngắt: Có một vấn đề rất thực tế đặt ra là nếu tại cùng một thời điểm có nhiều yêu cầu ngắt thuộc các loại ngắt khác nhau cùng đòi hỏi CPU phục vụ thì CPU sẽ phải có cơ chế để xử lý các yêu cầu ngắt này. Cơ chế phổ biến là chia các ngắt theo mức ưu tiên. CPU 8086 có khả năng phân biệt các mức ưu tiên khác nhau cho các loại ngắt (theo thứ tự từ cao xuống thấp) như sau:

+ ngắt trong: INT 0 (phép chia cho 0), INT N, INTO . . . cao nhất

+ ngắt không che được NMI

+ ngắt che được INTR

+ ngắt để chạy từng lệng INT 1 . . . thấp nhất

**4.Tập lệnh**

Bộ xử lý 8086 có tập lệnh gồm 111 lệnh, chiều dài của lệnh từ 1 byte đến vài byte. Gồm một số nhóm sau:

1. **Các lệnh trao đổi dữ liệu:** Các câu lệnh trong nhóm cho phép trao đổi dữ liệu giữa thanh ghi và ô nhớ hay giữa thiết bị vào/ra với ô nhớ hoặc thanh ghi. Kích cỡ dữ liệu cho phép với các câu lệnh này là byte (8 bít) hoặc word (16 bít). Như vậy các câu lệnh trao đổi dữ liệu giúp nạp dữ liệu cần thiết cho các thao tác tính toán của vi xử lý. Ngoài ra các lệnh này cho phép lưu các kết quả tính toán ra bộ nhớ hoặc các thiết bị ngoại vi.

|  |  |
| --- | --- |
| **Mã gợi nhớ** | **Chức năng** |
| **MOV** | Di chuyển byte hay word giữa thanh ghi và ô nhớ |
| **IN, OUT** | Đọc, ghi một byte hay word giữa cổng và ô nhớ |
| **LEA** | Nạp địa chỉ hiệu dụng |
| **PUSH, POP** | Nạp vào, lấy ra một word trong ngăn xếp. |
| **XCHG** | Hoán đổi byte hay word |

1. **Các lệnh tính toán số học và lô gíc:** Đây là các nhóm lệnh thực hiện các tính toán chủ yếu của vi xử lý 8086.

|  |  |
| --- | --- |
| **Mã gợi nhớ** | **Chức năng** |
| **NOT** | Đảo (bù một) byte hay word |
| **AND** | Phép và byte hoặc word |
| **OR** | Phép hoặc byte hoặc word |
| **XOR** | Phép hoặc loại trừ byte hoặc word |
| **SHL, SHR** | Dịch trái, dịch phải lôgíc byte hay word. Số bước 1 hoặc do CL xác định |
| **SAL, SAR** | Dịch trái, dịch phải số học byte hay word. Số bước 1 hoặc do CL xác định |
| **ROL, ROR** | Quay trái, quay phải byte hay word. Số bước 1 hoặc do CL xác định |
| **ADD, SUB** | Cộng trừ byte hoặc word |
| **ADC, SBB** | Cộng trừ byte hoặc word có nhớ |
| **INC, DEC** | Tăng, giảm |
| **NEG** | Đảo byte hoặc word (bù 2) |
| **CMP** | So sánh hai byte hoặc word |
| **MUL, DIV** | Nhân, chia byte hoặc word không dấu |
| **IMUL, IDIV** | Nhân chia byte hoặc word có dấu |

1. **Điều khiển, rẽ nhánh và lặp:** Các câu lệnh thuộc nhóm này cho phép thay đổi trật tự thực hiện các câu lệnh bên trong chương trình

|  |  |
| --- | --- |
| **Mã gợi nhớ** | **Chức năng** |
| **JMP** | Nhảy không điều kiện |
| **JA( JNBE)** | Nhảy nếu lớn hơn |
| **JAE( JNB)** | Nhảy nếu lớn hơn hoặc bằng |
| **JB(JNA)** | Nhảy nếu bé hơn |
| **JBE( JNA)** | Nhảy nếu bé hơn hoặc bằng |
| **JE(JZ)** | Nhảy nếu bằng |
| **JC, JNC** | Nhảy nếu cờ nhớ đặt, xóa |
| **JO, JNO** | Nhảy nếu cờ tràn đặt, xóa |
| **JS, JNS** | Nhảy nếu cờ dấu đặt, xóa |
| **LOOP** | Lặp không điều kiện, số lần lặp do CX xác định |
| **LOOPE(LOOPZ)** | Lặp nếu bằng (cờ không) hoặc số lần lặp do CX xác định |
| **LOOPPNE( LOOPNZ)** | Lặp nếu không bằng (cờ không xóa) hoặc số lần lặp do CX xác định |
| **CALL, RET** | Gọi hàm, trở về từ hàm con |
| **INT** | Ngắt mềm |
| **IRET** | Quay trở về từ đoạn chương trình ngắt |

1. **Điều khiển vi xử lý:** Các câu lệnh này tác động lên thanh ghi cờ là thay đổi trạng thái hoạt động của vi xử lý.

|  |  |
| --- | --- |
| **Mã gợi nhớ** | **Chức năng** |
| **STC, CLC, CMC** | Lập, xóa cờ nhớ |
| **STD, CLD** | Lập xóa cờ hướng |
| **STI, CLI** | Lập xóa cờ cho phép ngắt |
| **PUSHF, POPF** | Nạp vào, lấy ra thanh ghi cờ tới/từ ngăn xếp |
| **NOP** | Không làm gì cả |
| **WAIT** | Chờ tín hiệu TEST |
| **HLT** | Treo vi xử lý |

**5.Các chế độ địa chỉ**

* Chế độ địa chỉ (addressing mode) là cách để CPU tìm thấy toán hạng cho các lệnh của nó khi hoạt động. Một bộ vi xử lý có thể có nhiều chế độ địa chỉ. Các chế độ địa chỉ này được xác định ngay từ khi chế tạo ra bộ bi xử lý và sau này không thể thay đổi được.
* Bộ vi xử lý 8086 có 7 chế độ địa chỉ sau:

1. Chế độ địa chỉ thanh ghi (register addressing mode):

+ Người ta dùng các thanh ghi bên trong CPU như là các toán hạng để chứa dữ liệu cần thao tác. Vì vậy khi thực hiện lệnh có thể đạt tốc độ truy nhập cao hơn so với các lệnh có truy nhập đên bộ nhớ.

Ví dụ:

MOV BX, DX ; chuyển nội dung DX vào BX.

MOV DS, AX ; chuyển nội dung AX vào DX

1. Chế độ địa chỉ tức thì (immediate addressing mode):

+ Trong chế độ địa chỉ này, toán hạng đích là một thanh ghi hay một ô nhớ, còn toán hạng nguồn là một hằng số và vị trí của toán hạng này ở ngay sau mã lệnh. Chế độ địa chỉ này có thể được dùng để nạp dữ liệu cần thao tác vào bất kỳ thanh ghi nào (ngoại trừ các thanh ghi đoạn và thanh cờ) hoặc vào bất kỳ ô nhớ nào trong đoạn dữ liệu DS.

Ví dụ:

MOV CL, 100 ; chuyển 100 vào CL.

MOV AX, 0FF0H ; chuyển 0FF0H vào AX.

1. Chế độ địa chỉ trực tiếp (direct addressing mode):

+ Trong chế độ địa chỉ này một toán hạng chứa địa chỉ lệnh của ô nhớ dùng chứa dữ liệu còn toán hạng kia chỉ có thể là thanh ghi mà không được là ô nhớ. Nếu so sánh với chế độ địa chỉ tức thì ta thấy ở đây ngay sau mã lệnh không phải là toán hạng mà là địa chỉ lệch của toán hạng.

Ví dụ:

MOV AL, (1234H) ;chuyển ô nhớ DS:1234 vào AL.

MOV (4320H), CX ;chuyển CX vào 2 ô nhớ liên tiếp DS:4320 và DS:4321

1. Chế độ địa chỉ gián tiếp qua thanh ghi (register indirect addressing mode):

+ Trong chế độ địa chỉ này một toán hạng là một thanh ghi được sử dụng để chứa địa chỉ lệch của ô nhớ chứa dữ liệu, còn toán hạng kia chỉ có thể là thanh ghi mà không được là ô nhớ (8086 không cho phép tham chiếu bộ nhớ 2 lần đối với một lệnh).

Ví dụ:

MOV AL, (BX) ; chuyển ô nhớ có địa chỉ DS:BX vào AL.

MOV (SI), CL ; chuyển CL vào ô nhớ có địa chỉ DS:SI.

1. Chế độ địa chỉ tương đối cơ sở (based indexed relative addressing mode):

+ Trong chế độ địa chỉ này các thanh ghi cơ sở như BX và BP và các hằng số biểu diễn các giá trị dịch chuyển (displacement values) được dùng để tính địa chỉ hiệu dụng của toán hạng trong các vùng nhớ DS và SS. Sự có mặt của các giá trị dịch chuyển xác định tính tương đối của địa chỉ so với địa chỉ cơ sở.

Ví dụ:

MOV CX, (BX) +10 ; chuyển 2 ô nhớ liên tiếp có địa chỉ DS: (BX + 10) và ; DS: (BX+10) vào CX.

MOV AL, (BP) +5 ; chuyển ô nhớ SS: (BP+5) vào AL.

1. Chế độ địa chỉ tương đối chỉ số (indexed relative addressing mode):

+ Kết hợp hai chế độ địa chỉ chỉ số và cơ sở ta có chế độ địa chỉ chỉ số cơ sở. Trong chế độ địa chỉ này ta dùng cả thanh ghi cơ sở lẫn thanh ghi chỉ số để tính địa chỉ của toán hạng. Nếu ta dùng thêm cả thành phần biểu diễn sự dịch chuyển của địa chỉ thì ta có chế độ địa chỉ phức tạp nhất: chế độ địa chỉ tương đối chỉ số cơ sở. Ta có thể thấy chế độ địa chỉ này rất phù hợp cho việc địa chỉ hoá các mảng hai chiều.

Ví dụ:

MOV AX, [ BX ] [SI ]+8 ;chuyển 2 ô nhớ liên tiếp có địa chỉ ; DS:(BX+SI+8) và DS:(BX+SI+9) vào AX.

MOV AX, [BX+SI+8] ; một cách viết khác của lệnh trên MOV CL, [BP+DI+5 ; chuyển ô nhớ SS:(BP+DI+5) vào CL.

1. Chế độ địa chỉ tương đối chỉ số cơ sở (based indexed relative addressing mode):

+ Các thanh ghi đoạn và thanh ghi lệch được ngầm định đi kèm với nhau từng cặp dùng để địa chỉ hoá các toán hạng trong các vùng khác nhau của bộ nhớ. Vì tính ngầm định nên trong các lệnh ta chỉ cần viểt các thanh ghi lệch là đủ cơ sở để tính ra được đia chỉ của toán hạng. Tuy nhiên, ngoài các tổ hợp ngầm định đã kể, 8086 còn cho phép ta làm việc với các tổ hợp ngầm định đã kể, 8086 còn cho phép ta làm việc với các tổ hợp khác của các thanh ghi đoạn và thanh ghi lệch. Muốn loại bỏ các tổ hợp ngầm định nói trên, trong khi viết lệnh phải ghi rõ thanh ghi đoạn sẽ dùng để tính địa chỉ.

Các cặp thanh ghi đoạn và thanh ghi lệch ngầm định:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thanh ghi đoạn | CS | DS | ES | SS |
| Thanh ghi lệch | IP | SI, DI, BX | DI | SP, BP |

Ví dụ: Nếu ta muốn thay đổi, không lấy toán hạng trong đoạn dữ liệu DS, mà lại lấy toán hạng trong đoạn dữ liệu phụ ES để đưa vào AL, thì ta phải viết lại lệnh trên thành: MOV AL, ES:[BX].

Trong đó ta đã dùng ES: để loại bỏ thanh ghi đoạn ngầm định DS và để chỉ rõ thanh ghi đoạn mới dùng trong lệnh này bây giờ là ES.

**III. THIẾT KẾ HỆ THỐNG**

**1. Cấu trúc hệ thống**

Cấu trúc hệ thống báo cháy báo cháy bao gồm:

- Cảm biến nhiệt độ DS18B20

- Cảm biến khói sử dụng module cảm biến khí gas MQ2.

- Dùng bộ điều khiển với bộ xử lí trung tâm là vi xử lí 8086 để giao tiếp với cảm biến nhiệt độ, cảm biến khói và phát tín hiệu đến các thiết bị báo động

- Thiết bị báo động khi có hỏa hoạn là chuôg và đèn báo.

**2. Sơ đồ khối hệ thống**

KHỐI NGUỒN

KHỐI PHÁT TÍN HIỆU

KHỐI XỬ LÝ

KHỐI CẢM BIẾN KHÓI

KHỐI CẢM BIẾN NHIỆT ĐỘ

-Khối cảm biến nhiệt độ: Sử dụng cảm biến nhiệt độ DS18b20 có nhiệm vụ lấy nhiệt độ môi trường xung quanh để đưa thông tin về khối xử lý và đưa ra cảnh báo khi nhiệt độ vượt qua ngưỡng cho phép.

-Khối cảm biến khói: Sử dụng Module cảm biến khói MQ2 có nhiệm vụ phát hiện khói và đưa thông tin về khối xử lý.

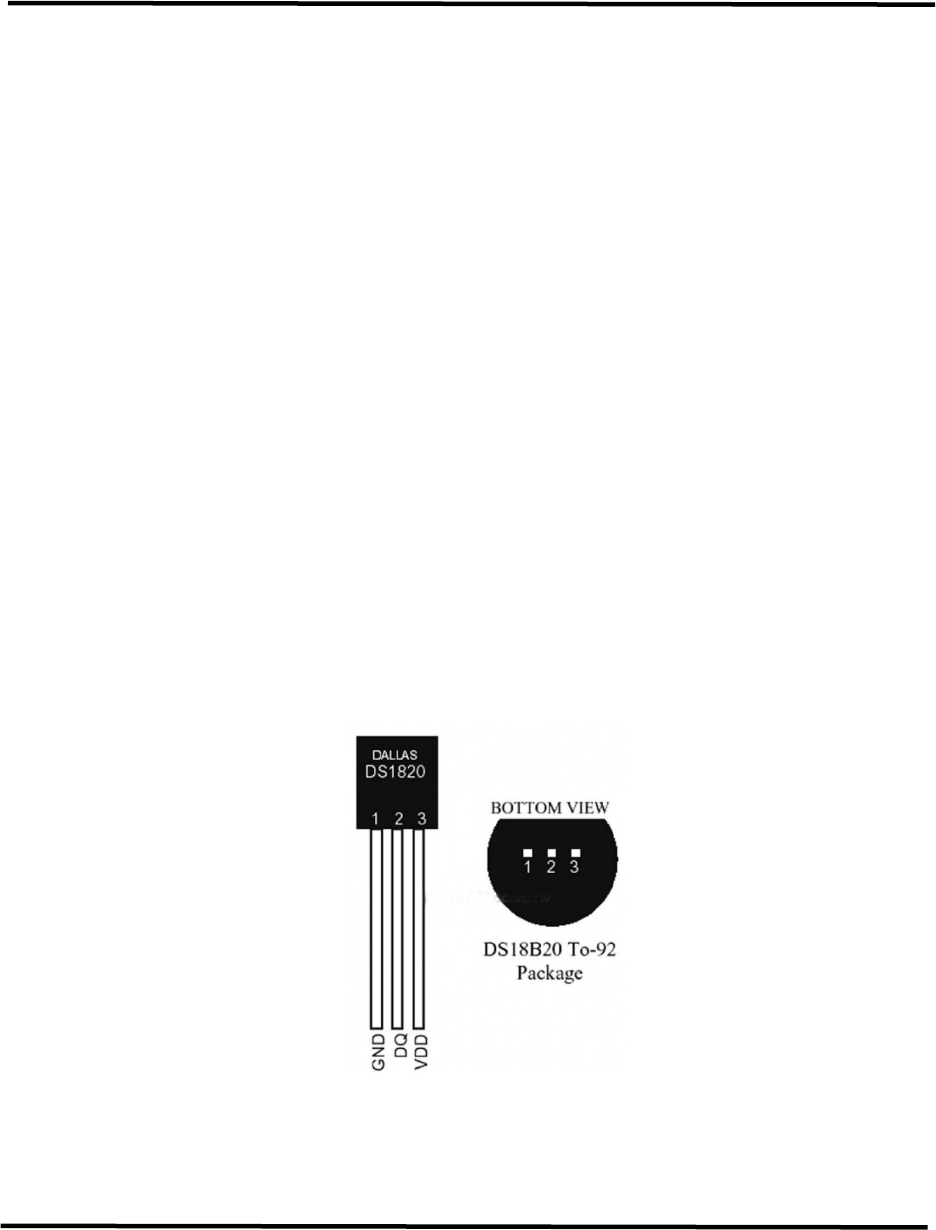
-Khối xử lý: Sử dụng khối điều khiển với bộ xử lý trung tâm là vxl 8086 dùng để giao tiếp với các khối cảm biến nhiệt độ cảm biến khói qua cổng P1.0 và P2.0 và đưa thông tin đến khối phát tín hiệu ra bằng cổng P1.6 và P1.7.

-Khối phát tín hiệu: Sử dụng còi báo và đèn có nhiệm vụ báo động cho mọi người biết khi có cháy xảy ra.

-Khối nguồn: Sử dụng nguồn điện DC có điện áp 5V để cung cấp cho hoạt động của mạch.

**3. Cấu tạo và nguyên lý hoạt động của các linh kiện**

a. Cảm biến nhiệt độ DS18B20



Cảm biến nhiệt độ DS18B20

Các chân linh kiện:

-VCC: Chân cấp nguồn cho cảm biến, điện áp sử dụng 5V

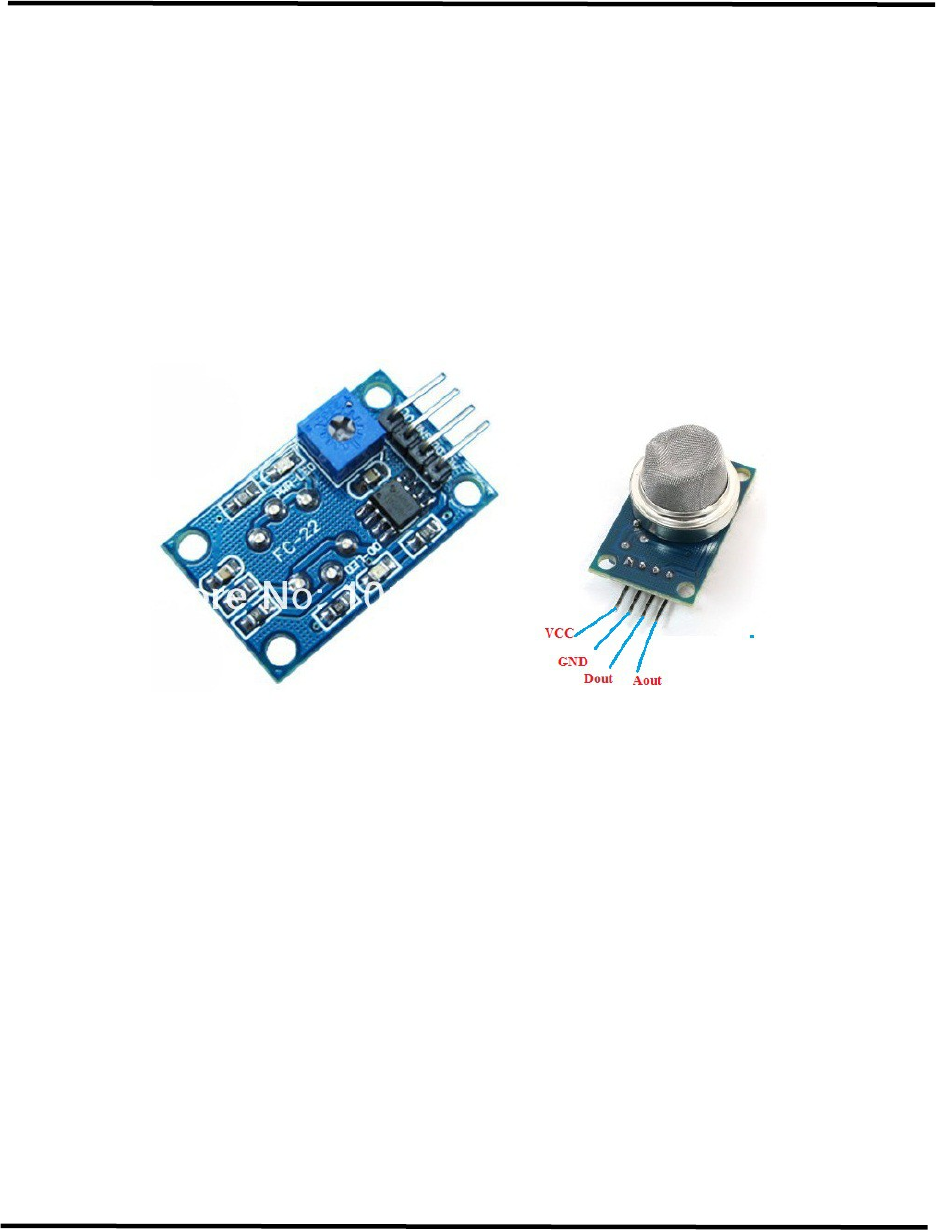
-GND: Chân nối đất cho cảm biến

-DQ: Chân dữ liệu ra của cảm biến

Cảm biến DS18B20  đo nhiệt độ theo mức một wire, lấy nhiệt độ với độ phân giải có cấu hình từ 9 đến 12 bit. Nếu không cấu hình sẽ để mặc định ở 12 bit. Nhiệt độ đo được từ -55⁰C tới 125⁰C. Có thể lập trình mức nhiệt độ cảnh báo khi vượt qua nguồn cho phép.

b. Cảm biến khói

Module cảm biến khí ga MQ2:



Module cảm biến khí GAS MQ2

Sơ lược về cảm biến:

-VCC: Chân cấp nguồn cho cảm biến, điện áp sử dụng 5V

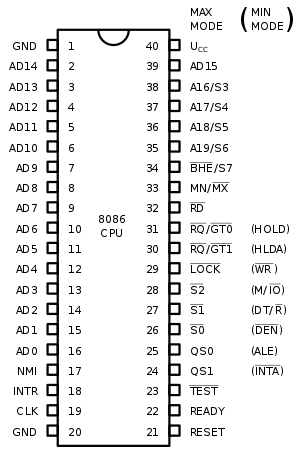
-GND: Chân nối đất cho cảm biến

-AOUT: Đầu ra ADC

-DOUT: Đầu ra Digital ( So sánh khi phát hiện khí)

Module MQ2 là thiết bị dùng để cảm biến khí, trong đó có tác dụng cảm biến khói. Khi không có khói DOUT của cảm biến đưa ra giá trị ở mức cao, khi có khói thì DOUT của cảm biến đưa ra giá trị ở mức thấp.

c.Vi xử lý 8086



Sơ đồ chân tín hiệu 8086

Một số chân được sử dụng trong mạch:

-VCC: Chân cấp nguồn , điện áp sử dụng 5V

-GND: Chân nối đất , điểm 0V của nguồn nuôi

-RST: Chân Reset

-AD0 – AD15 [I/O: tín hiệu vào và ra]: Các chân dồn kênh cho các tín hiệu buýt

dữ liệu và buýt địa chỉ. Xung ALE sẽ báo cho mạnh ngoài biết khi nào trên các

đường đó có tín hiệu dữ liệu (ALE = 0) hoặc địa chỉ (ALE = 1). Các chân này ở

trạng thái trở kháng cao khi muyP chấp nhận treo.

- A16/S3, A17/S4, A18/S5, A19/S6 [O]: Các chân dồn kênh của địa chỉ phần cao

và trạng thái. Địa chỉ A16 - A19 được truyền trên các chân đó khi ALE = 1 còn

khi AEL = 0 thì trên các chân đó có các tín hiệu trạng thái S3-S6. Các chân này ở

trạng thái trở kháng cao khi muyP chấp nhận treo.

- RD[O]: Xung cho phép đọc. Khi RD= 0 thì buýt dữ liệu sẵn sàng nhận số liệu

từ bộ nhớ hoặc thiết bị ngoại vi. Chân RD ở trạng thái trở kháng cao khi muyP

chấp nhận treo.

- READY [I]: Tín hiệu báo cho CPU biết tình trạng sẵn sàng của thiết bị ngoại vi

hay bộ nhớ. Khi READY = 1 thì CPU thực ghi/đọc mà không cần chèn thêm các

chu kỳ đợi. Ngược lại khi thiết bị ngoại vi hay bộ nhớ có tốc độ hoạt động chậm,

chúng có thể đưa tín hiệu READY = 0 để báo cho CPU biết. Lúc này CPU tự kéo

dài thời gian thực hiện lệnh ghi/đọc bằng cách chèn thêm các chu kỳ đợi.

- INTR [I]: Tín hiệu yêu cầu ngắt che được. Khi có yêu cầu ngắt mà cờ cho phép

ngắt IF = 1 thì CPU kết thúc lệnh đang làm dở, sau đó nó đi vào chu kỳ chấp nhận

ngắt và đưa ra bên ngoài tín hiệu INTA = 0.

-TEST[I]: Tín hiệu tại chân này được kiểm tra bởi lệnh WAIT. Khi CPU thực

hiện lệnh WAIT mà lúc đó tín hiệu TEST=1, nó sẽ chờ cho đến khi tín hiệu

TEST= 0 thì mới thực hiện lệnh tiếp theo.

- NMI [I]: Tín hiệu yêu cầu ngắt không che được. Tín hiệu này không bị khống

chế bởi cờ IF và tín hiệu này sẽ được CPU nhận biết bằng các tác động của sườn

lên của xung yêu cầu ngắt. Nhận được yêu cầu này CPU kết thúc lệnh đang làm

dở, sau đó chuyển sang thực hiện chương trình phục vụ ngắt kiểu INT2.

- RESET [I]: tín hiệu khởi động lại 8086. khi RESET = 1 kéo dài ít nhất trong thời

gian 4 chu kỳ đồng hồ thì 8086 bị buộc phải khởi động lại: nó xoá các thanh ghi

DS, ES, SS, IP và FR về 0 và bắt đầu thực hiện chương trình tại địa chỉ

CS:IP=FFFF:0000H (chú ý cờ IF 0 để cấm các yêu cầu ngắt khác tác động vào

CPU và cờ TF=0 để bộ vi xử lý không ở chế độ chạy từng lệnh).

- CLK [I]: Tín hiệu đồng hồ (xung nhịp).cung cấp xung nhịp làm việc cho CPU.

- MN/MX [I]: Chân điều khiển hoạt động của CPU theo chế độ MIN/MAX.

- IO/M[O] khi ở chế độ MIN: Tín hiệu này phân biệt trong thời điểm đã định phần tử nào trong các thiết bị vào/ra (IO) hoặc bộ nhớ (M) được chọn làm việc với

CPU. Trên buýt địa chỉ lúc đó sẽ có các địa chỉ tương ứng của các thiết bị

đó. Chân này ở trạng thái trở kháng cao khi muyP chấp nhận treo.

-WR[O] khi ở chế độ MIN: Xung cho phép ghi. Khi CPU đưa ra WR=0 thì trên buýt dữ liệu các dữ liệu đã ổn định và chúng sẽ được ghi vào bộ nhớ hoặc thiết bị

ngoại vi tại thời điểm WR= 1. ChânWR ở trạng thái trở kháng cao khi muyP

chấp nhận treo.

- INTA [O]: Tín hiệu báo cho các mạch bên ngoài biết CPU chấp nhận yêu

cầu ngắt INTR. Lúc này CPU đưa ra INTA = 0 để báo là nó đang chờ

mạch ngoài đưa vào số hiệu ngắt (kiểu ngắt) trên buýt dữ liệu.

- ALE [O]: Xung cho phép chốt địa chỉ. Khi ALE = 1 có nghĩa là trên buýt

dồn kênh AD có các địa chỉ của thiết bị vào/ra hay của ô nhớ. ALE không

bao giờ bị thả nối (trong trạng thái trở kháng cao) khi CPU bị treo thì

ALE= 0.

- DT/R[O]: Tín hiệu điều khiển các đệm 2 chiều của buýt dữ liệu để chọn

chiều chuyển của vận dữ liệu trên buýt D. Chân này ở trạng thái trở kháng

cao khi muyP chấp nhận treo.

-DEN[O]: Tín hiệu báo cho bên ngoài biết là lúc này trên buýt dồn kênh

AD có dữ liệu ổn định. Chân này ở trạng thái trở kháng cao khi muyP chấp

nhận treo.

-HOLD [I]: Tín hiệu yêu cầu treo CPU để mạch ngoài thực hiện việc trao

đổi dữ liệu với bộ nhớ bằng cách truy nhập trực tiếp. Khi HOLD = 1. CPU

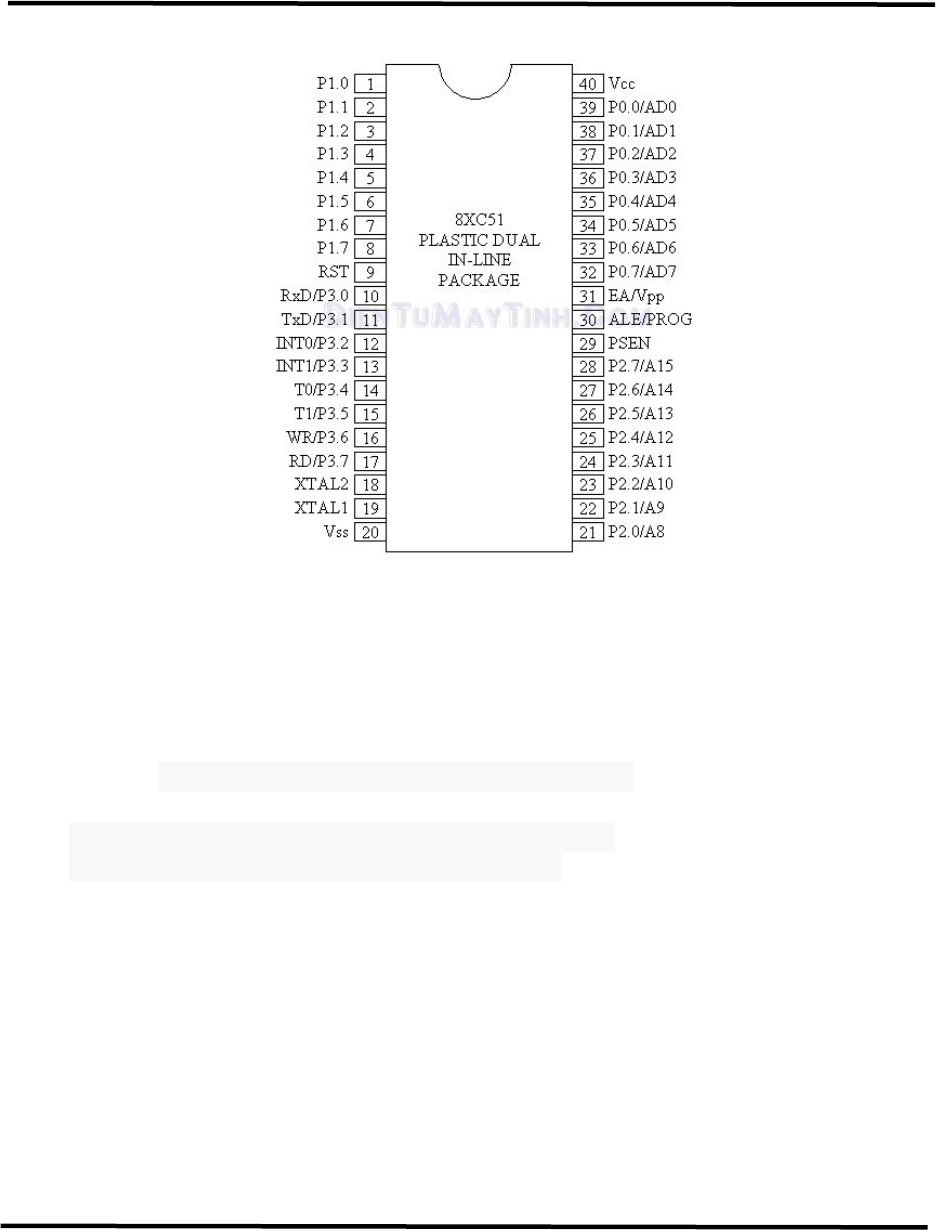
8086 sẽ tự tách ra hệ thống bằng cách treo tất cả các buýt A, buýt D, buýt

C (các buýt ở trạng thái trở kháng cao) để bộ điều khiển DMA (DMA

Contrroller) lấy quyền điều khiển hệ thống để thực hiện trao đổi dữ liệu.

-….

d.Khối điều khiển với bộ xử lí trung tâm là vxl 8086



Khối điều khiển

Sơ đồ chân IC khối điều khiển

Một số chân được sử dụng trong mạch:

-VCC: Chân cấp nguồn cho IC, điện áp sử dụng 5V

-GND: Chân nối đất cho IC

-EA/VPP:Cho phép làm việc với bộ nhớ chương trình

-RST: Chân Reset

-XTAL1,XTAL2: 2 chân tạo dao động clock cho IC

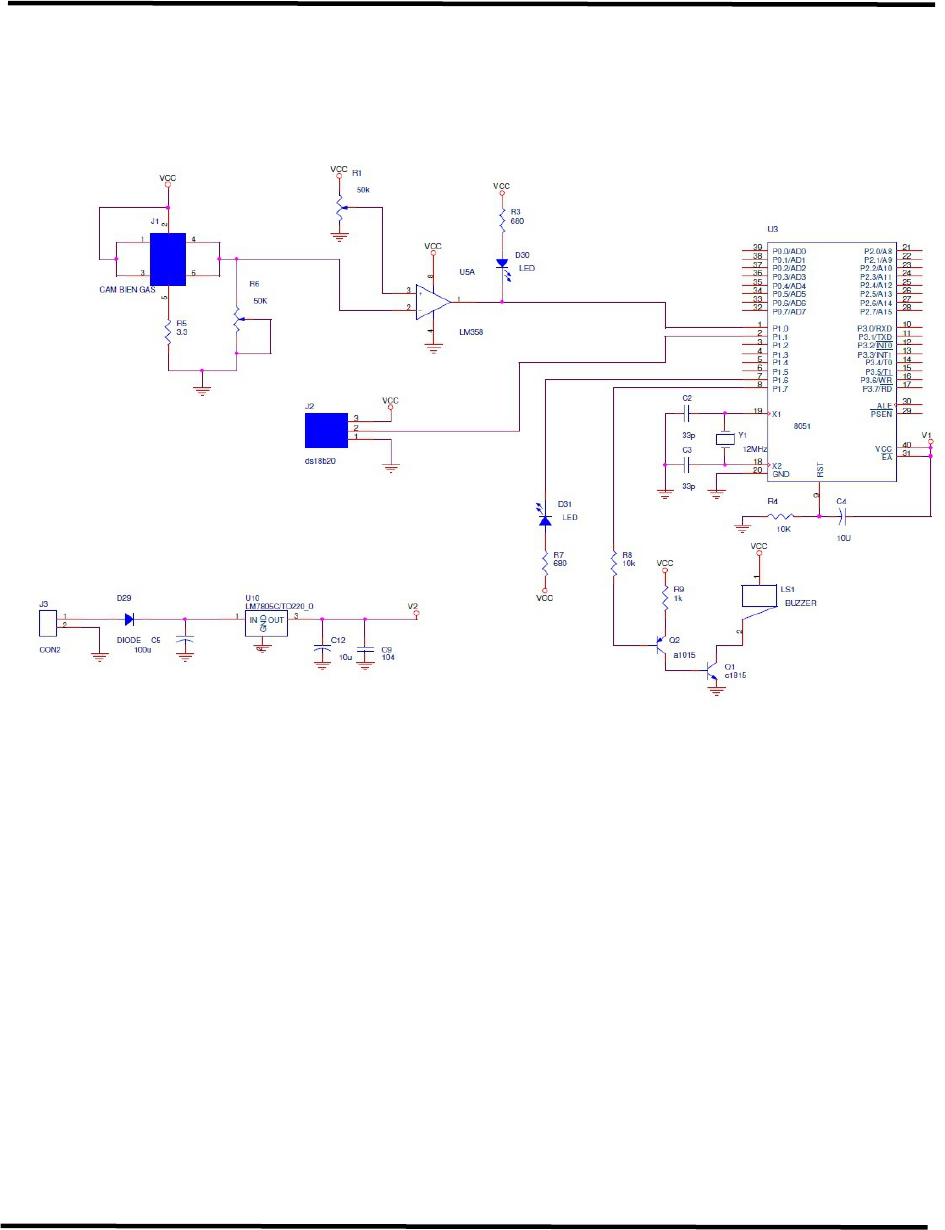
-P1.0,P1.1,P1.6 và P1.7: Các chân của cổng 1

**4.Nguyên lý hoạt động và sơ đồ nguyên lý**

a. Nguyên lý hoạt động

Khi có cháy, nhiệt độ môi trường xung quanh cao lên đến khi vượt qua ngưỡng giới hạn của cảm biến nhiệt độ DS18B20 (>50℃) đưa ra tín hiệu về bộ xử lý. Và khi mật độ khói tăng lên, cảm biến khói đang có đầu ra là mức cao( không có khói) sẽ nhảy sang mức thấp (có khói).Khối điều khiển với bộ xử lý trung tâm (vxl 8086) có nhiệm vụ xử lý dữ liệu cảm biến truyền qua. Sau khi xử lý xong, bộ xử lý trung tâm sẽ đưa dữ liệu đã xử lý và phát tín hiệu qua chuông và đèn báo động.

b.Sơ đồ nguyên lý



**IV.THUẬT TOÁN**

**1. Lưu đồ giải thuật**

I=CẢM BIẾN NHIỆT ĐỘ

J=CẢM BIẾN KHÓI

S

S

I>50

J=0

Đ

Đ

Xử lý tín hiệu

Đèn báo sáng

Chuông báo kêu

**2.Giải thích lưu đồ**

Lưu đồ trên bắt đầu,ta gán I là cảm biến nhiệt độ, J là cảm biến khói.Tiếp theo ta so sánh I>50°C.Khi đó,ta có 2 trường hợp xảy ra:

-Trường hợp 1: I<=50°C, chương trình quay lại bước gán.

-Trường hợp 2:I>50°C, chương trình đưa dữ liệu về bộ xử lý.

Tiếp tục chương trình,ta xem đầu ra của cảm biến khói J có ở mức thấp hay không,ta có 2 trường hợp xảy ra:

-Trường hợp 1:J=1,đầu ra của cảm biến khói đang ở mức cao,nghĩa là không có khói nên chương trình quay lại bước gán.

-Trường hợp 2:J=0,đầu ra của cảm biến khói đang ở mức thấp,nghĩa là có khói,chương trình đưa dữ liệu về bộ xử lý.

Khi nhận được dữ liệu từ khối cảm biến nhiệt độ,lúc này bộ xử lý sẽ phát tín hiệu cho đèn báo động. Và đồng thời nhận được dữ liệu từ cảm biến khói đưa về thì chuông báo động sẽ kêu lên.