1. Các tín hiệu điều khiển điển hình:

MEMR (Memory Read): điều khiển đọc lệnh/dữ liệu từ một ngăn nhớ có địa chỉ xác định.

MEMW (Memory Write): điều khiển ghi dữ liệu đến một ngăn nhớ có địa chỉ xác định.

IOR (I/O Read): điều khiển đọc dữ liệu từ một cổng vào ra có địa chỉ xác định.

IOW (I/O Write): điều khiển ghi dữ liệu đến một cổng vào ra có địa chỉ xác định.

INTR (Interrupt Request): tín hiệu từ mạch điều khiển vào ra gửi đến bộ xử lý yêu cầu ngắt (ngắt chắn được).

INTA (Interrupt Acknowledge): tín hiệu từ bộ xử lý gửi ra ngoài báo chấp nhận ngắt.

NMI (Non Maskable Interrupt): tín hiệu ngắt gửi đến bộ xử lý (ngắt không chắn được).

Reset: tín hiệu từ bên ngoài khởi động lại máy tính.

Hold/Bus Repuest: tins hiệu từ mạch điều khiển vào ra gửi đến bộ xử lý yêu cầu nhường quyền diều khiển bus.

HLDA/Bus Grant: tín hiệu phát ra từ bộ xử lý chấp nhận nhường quyền điều khiển bus.

Chú thích:

- Tín hiệu điều khiển đọc/ghi.

- Tín hiệu điều khiển ngắt.

Ngắt là cơ chế cho phép bộ xử lý tạm thời dừng thực hiện công việc hiện tại để chuyển qua thực hiện công việc khác, gọi là chương trình con phục vụ ngắt. có 3 loại ngắt đó là ngắt cứng ngắt mềm và ngắt ngoại lệ.

Chu trình ngắt được thêm vào chu trình lệnh. Bộ xử lý kiểm tra tín hiệu ngắt sau khi hoàn thành mỗi lệnh nếu không có ngắt bộ xử lý nhận lệnh kế tiếp. nếu có ngắt (không bị cấm) bộ xử lý thực hiện: 1. Tạm dừng thực hiện chương trình hiện tại 2. Cất trạng thái xử lý hiện tại. 3. Xử lý ngắt. 4. Khôi phục trạng thái và tiếp tục chương trình đang bị ngắt.

- Tín hiệu điều khiển bus.

1. Luật Moore

* Số lượng transistor tăng gấp đôi sau mỗi 18 tháng.
* Giá thành của chip hầu như không thay đổi.
* Mật độ cao hơn => đường dẫn điện ngắn hơn, dẫn đến hiệu năng cao hơn.
* Kích thước nhỏ hơn => độ phức tạp tăng lên.
* Công suất tiêu thụ nhỏ hơn và phải làm mát.
* Hệ thống có ít chip liên kết với nhau, do đó tăng độ tin cậy

1. Biều diễn số nguyên không dấu và số nguyên có đấu

* Số nguyên không dấu: dùng n chữ số nhị phân thì biểu diễn được 2n số. Dải biểu diễn từ 0 => 2n – 1.
* Số nguyên có dấu: dùng n chữ số nhị phân thì biểu diễn được 2n số. Dải biểu diễn -2n-1 => 2n-1 – 1.

Để biểu diễn số nguyên có dấu ta có 2 cách đó là dùng dấu hoặc biểu diễn dạng mã bù 2. Đối với cách dùng dấu thì bit đầu tiên biểu diễn dầu (0 biểu diễn cho số dương và 1 biểu diễn cho số âm) còn các bit còn lại biểu diễn giá trị. Ví dụ: 23 = 00010111 còn -23 = 10010111. Cách dùng mã bù 2. Đối với số dương thì biểu diễn dạng nhị phân bình thường còn đối với số âm bước 1 ta đổi số dương tương ứng sang nhị phân. Sau đó tìm số bù 1 bằng cách đảo bit 0 => 1 và 1 => 0. Cộng 1 vào số bù 1 ta sẽ tìm được số bù 2.

Cách chuyển số âm từ mã bù 2 là đặt dấu trừ vào bit đầu tiên vầ cộng với các bit còn lại.

* Chuyển đổi độ dài số dương thì thêm bớt số 0 ở bên trước còn đối với số âm thì thêm bớt số 1 ở bên trước.
* Khi cộng hai số nguyên không dấu n bit: nếu không có nhớ ra bit cao nhất thì tổng luôn đúng. Nếu có nhớ ra ngoài thì là tổng sai, ta nói rằng phép cộng đã tràn nhớ.
* Khi cộng hai số nguyên có dấu n bit, không quan tâm đến bit nhớ ra ngoài, kết quả nhận được là n bit: nếu cộng hai số khác dấu, tổng thu được luôn đúng. Cộng hai số nguyên cùng dấu, nếu tổng cùng dấu với các số hạng thì tổng đó đúng. Cộng hai số cùng dấu nếu tổng ngược dấu với các số hạn thì tổng đó sai.

1. Chuyển đổi số nguyên sang dạng nhị phân chuẩn IEE 754/85 (32 bit 8, 64 bit 11)

* Bước 1: Đổi số thực X đó qua dạng nhị phân
* Bước 2: Chuyển về dạng 1,aaa…a\*2b
* Bước 3:
  + if X > 0 S = 0 else S = 1
  + E -127 = b => E = 127 + b đổi E sang nhị phân (8 bit)
  + M = aaa…a0000 đủ 23 bit
* X = SEM đủ 32 bit.

1. Phân tích các nhiệm vụ của bộ xử lý.

* Nhận lệnh: Nhận lệnh từ bộ nhớ. Bên trong bộ xử lý có hai thanh ghi liên quan trực tiếp đến quá trình nhận lệnh:

+ Thanh ghi bộ đếm chương trình: chứa địa chỉ của lệnh sẽ được nhận vào.

+ Thanh ghi lệnh: chứa lệnh được nhận về.

Từ địa chỉ của lệnh cần thực hiện nằm trong bộ đếm chương trình (PC), bộ xử lý tìm ra ngăn nhớ chứa lệnh.

Tiếp theo, bộ xử lý phát ra tín hiệu đọc ngăn nhớ vừa tìm được.

Nội dung của ngăn nhớ được chuyển qua bus dữ liệu và đưa đến thanh ghi lệnh (IR).

* Giải mã lệnh: giải mã để biết lệnh làm gì. Lệnh từ thanh ghi lệnh được đưa đến khối điều khiển. Tại đây, lệnh được giải mã để xác định thao tác mà lệnh yêu cầu. Khi đó khối điều khiển sẽ phát ra tín hiệu điều khiển tương ứng với lệnh đó.
* Nhận dữ liệu: Lệnh có thể yêu cầu nhận dữ liệu từ bên ngoài vào. Bộ xử lý phát ra địa chỉ của ngăn nhớ hoặc cổng vào ra chứa dữ liệu cần nhận. Bộ xử lý phát ra tín hiệu đọc ngăn nhớ hoặc cổng vào ra tương ứng. Dữ liệu được chuyển qua bus dữ liệu và đưa vào tập thanh ghi bên trong.
* Xử lý dữ liệu: Lệnh có thể yêu cầu thực hiện một phép toán nào đó. Dữ liệu được chuyển từ các thanh ghi vào ALU. ALU thực hiện các phép toán dưới sự điều khiển của khối điều khiển. Kết quả của phép toán được cất tạm thời vào thanh ghi dữ liệu.
* Ghi dữ liệu: Lệnh có thể yêu cầu cất kết quả ra ngoài. Bộ xử lý phát ra địa chỉ ngăn nhớ hoặc cổng vào ra cần ghi dữ liệu. Bộ xử lý phát ra tín hiệu điều khiển ghi ngăn nhớ hoặc cổng vào ra. Dữ liệu từ thanh ghi dữ liệu được chuyển ra ngoài.

Ngoài ra, trong quá trình thực hiện chương trình còn nhận các yêu cầu từ bên ngoài, xử lý các yêu cầu đó.

1. Các lệnh dữ liệu.

* Move nguồn, đích: Copy dữ liệu từ nguồn đến đích.
* Load: copy dữ liệu từ bộ nhớ đến bộ xử lý
* Store: copy dữ liệu từ bộ xử lý đến bộ nhớ.
* Exchange: tráo đổi nội dung của nguồn và đích
* Push: chuyển dữ liệu từ nguồn đến ngăn xếp
* Pop: chuyển dữ liệu từ ngăn xếp đến đích
* Add đích, nguồn: Cộng đích và nguồn kết quả chứa trong đích
* Subtract đích, nguồn: tính hiệu đích và nguồn kết quả chứa trong đích
* Multiply thừa số 1: lệnh này dùng để nhân các số nguyên không dấu. nếu thừa số một có kích thước 1 bit thì thừa số 2 sẽ là thanh ghi AL. kết quả chứa trong thanh ghi AX. Nếu thừa số 1 có kích thước 1 word thì thừa số 2 sẽ là thanh ghi AX. Kết quả của phép nhân được chứa trong hai thanh ghi DX:AX
* Divide số chia: tính thương 2 toán hạng. Nếu số chia có kích thước 1byte thì số bị chia sẽ chứa trong thanh ghi AX thương số sẽ chứa trong thanh ghi AL. số dư chứa trong AH. Nếu số chia có kích thước 2 byte thì số bị chia chứa trong DX:AX thương số chứa trong AX số dư chứa trong DX.
* Absolute: Lấy giá trị tuyệt đối của toán hạng
* Negate (NEG): Đổi dấu toán hạng (lấy bù 2)
* Increment (INC): tăng toán hạng lên 1
* Decrement (DEC): giảm toán hạng đi 1
* And đích, nguồn: thực hiện and bit 2 toán hạng giống
* Or đích, nguồn: thực hiện Or bit 2 toán hạng
* Xor đích, nguồn: thưc hiện Xor bit 2 toán hạng (giống thì là 0)
* Not đích: đảo bit của toán hạng (lấy bù 1)
* Test: thực hiện and 2 toán hạng để lập cờ
* Shift left: thực hiện dịch trái chèn 0 vào bên phải
* Shift right: thực hiện dịch phải chèn 0 vào bên trái
* Rotate left: thực hiện quay trái bit toán hạng
* Rotate right: thực hiện quay phải bit toán hạng
* Compare: so sánh 2 toán hạng

1. Các chế độ địa chỉ

* Chế độ tức thì: toán hạng nằm ngay trong trường địa chỉ ADD AX, 10
* Chế độ trực tiếp: Toán hạng là nội dung ngăn nhớ có địa chỉ được chỉ ra trong lệnh ADD AL, [100]
* Chế độ gián tiếp: Toán hạng là nội dung của ngăn nhớ có địa chỉ nằm trong một ngăn nhớ khác ADD AH, [a]
* Chế độ thanh ghi: Toán hạng là nội dung của thanh ghi ADD AL, AH
* Chế độ gián tiếp qua thanh ghi: Toán hạng là nội dung ngăn nhớ có địa chỉ nằm trong một thanh ghi ADD AL, [BX]
* Chế độ dịch chuyển: Là kết hợp chế độ gián tiếp qua thanh ghi và chế độ trực tiếp ADD AL, [SL + 50]
* Chế độ ngăn xếp toán hạng ngầm hiểu là đỉnh ngăn xếp PUSH AX

1. Các kỹ thuật ánh xạ địa chỉ cache

* Ánh xạ trực tiếp: mỗi block của bộ nhớ chính chỉ được ánh xạ vào một line duy nhất trong cache i = j mod m i số hiệu line trong cache, j số hiệu block trong bộ nhớ chính, m số lượng line trong cache.

Khi đó cpu phát ra 3 trường gồm word xác định số hiệu ngăn nhớ trong block => Block (line) có 2w ngăn nhớ

Line xác định số hiệu line trong cache => Cache có 2r line, cache chứa 2r + w ngăn nhớ.

Tag xác định block nào đang ở trong line => Bộ nhớ chính có chứa 2s+w ngăn nhớ.

A screenshot of a computer

Description automatically generated

* Ánh xạ liên kết hoàn toàn: Mỗi block trong bộ nhớ chính được ánh xạ vào một line bất kỳ trong cache khi đó địa chỉ do bộ xử lý phát ra có dạng

A screenshot of a computer

Description automatically generated

* Ánh xạ liên kết tập hợp

A white background with black text

Description automatically generatedA screenshot of a computer

Description automatically generated