|  |  |  |  |
| --- | --- | --- | --- |
| D2 | D1 | D0 | Mạch điều khiển |
| 0 | 0 | 0 | Mov A, C: đưa dữ liệu từ thanh ghi lưu kết quả C vào thanh ghi A |
| 0 | 0 | 1 | Mov B, C: đưa dữ liệu từ thanh ghi lưu kết quả C vào thanh ghi B |
| 0 | 1 | 0 | ADD (+) |
| 0 | 1 | 1 | SUB (-) |
| 1 | 0 | 0 | AND |
| 1 | 0 | 1 | OR |
| 1 | 1 | 0 | LOAD A: Đưa dữ liệu vào thanh ghi A |
| 1 | 1 | 1 | LOAD B: Đưa dữ liệu vào thanh ghi B |

VD: Mã hóa các lệnh thực hiện phép tính 5 + 2 cho vi xử lý đơn giản.

|  |  |
| --- | --- |
| Mảng thanh ghi | Tác động |
| 1 1 0 | Nạp thanh ghi A (đưa giá trị cần thực hiện phép tính vào thanh ghi A) = LOAD A |
| 1 0 1 | Với giá trị = 5 (101) |
| 1 1 1 | Nạp thanh ghi B (đưa giá trị cần thực hiện phép tính vào thanh ghi B) = LOAD B |
| 0 1 0 | Với giá trị = 2 (010) |
| 0 1 0 | Thực hiện phép tính **cộng** giữa thanh ghi A và B (A + B) và ghi kết quả vào thanh ghi C |

VD: Mã hóa các lệnh thực hiện phép tính (2 or 1 and 5) + 5 -2

Tính như bình thường, trong ngoặc trước, từ trái sang phải

|  |  |
| --- | --- |
| Mảng thanh ghi | Tác động |
| 1 1 0 | Nạp thanh ghi A (đưa giá trị cần thực hiện phép tính vào thanh ghi A) = LOAD A |
| 0 1 0 | Với giá trị = 2(0 1 0) |
| 1 1 1 | Nạp thanh ghi B (đưa giá trị cần thực hiện phép tính vào thanh ghi B) = LOAD B |
| 0 0 1 | Với giá trị = 1 (0 0 1) |
| 1 0 1 | Thực hiện phép tính **OR** giữa thanh ghi A và B (A or B) và ghi kết quả vào thanh ghi C  C = A or B= 2 or 1  A và B không còn dữ liệu gì cả |
| 0 0 0 | Đưa kết quả ở thanh ghi C vào thanh ghi A = MOV A, C  A= 2 or 1 |
| 1 1 1 | Nạp thanh ghi B (đưa giá trị cần thực hiện phép tính vào thanh ghi B) = LOAD B |
| 1 0 1 | Với giá trị = 5 (1 0 1) |
| 1 0 0 | Thực hiện phép tính **and** giữa thanh ghi A và B (A and B) và ghi kết quả vào thanh ghi C  C = A and B= 2 or 1 and 5  A và B không còn dữ liệu gì cả |
| 0 0 0 | Đưa kết quả ở thanh ghi C vào thanh ghi A = MOV A, C  A= 2 or 1 and 5 |
| 1 1 1 | Nạp thanh ghi B (đưa giá trị cần thực hiện phép tính vào thanh ghi B) = LOAD B |
| 1 0 1 | Với giá trị = 5 (1 0 1) |
| 0 1 0 | Thực hiện phép tính **cộng** giữa thanh ghi A và B (A + B) và ghi kết quả vào thanh ghi C  C= (2 or 1 and 5) + 5 |
| 0 0 0 | Đưa kết quả ở thanh ghi C vào thanh ghi A = MOV A, C  A= (2 or 1 and 5) + 5 |
| 1 1 1 | Nạp thanh ghi B (đưa giá trị cần thực hiện phép tính vào thanh ghi B) = LOAD B |
| 0 1 0 | Với giá trị = 2 (0 1 0) |
| 0 1 1 | Thực hiện phép tính **trừ** giữa thanh ghi A và B (A - B) và ghi kết quả vào thanh ghi C  C= (2 or 1 and 5) + 5 - 2 |

Vd: 3 and 4 or 6

|  |  |
| --- | --- |
| Mảng thanh ghi | Tác động |
| 1 1 0 | Nạp thanh ghi A (đưa giá trị cần thực hiện phép tính vào thanh ghi A) = LOAD A |
| 0 1 1 | Với giá trị = 3 (0 1 1) |
| 1 1 1 | Nạp thanh ghi B (đưa giá trị cần thực hiện phép tính vào thanh ghi B) = LOAD B |
| 1 0 0 | Với giá trị = 4 (1 0 0) |
| 1 0 0 | Thực hiện phép tính **and** giữa thanh ghi A và B (A and B) và ghi kết quả vào thanh ghi C  C = A and B= 3 and 4  A và B không còn dữ liệu gì cả |
| 0 0 0 | Đưa kết quả ở thanh ghi C vào thanh ghi A = MOV A, C  A= 3 and 4 |
| 1 1 1 | Nạp thanh ghi B (đưa giá trị cần thực hiện phép tính vào thanh ghi B) = LOAD B |
| 1 1 0 | Với giá trị = 6 (1 1 0) |
| 1 0 1 | Thực hiện phép tính **OR** giữa thanh ghi A và B (A or B) và ghi kết quả vào thanh ghi C  C = A or B= 3 and 4 or 6  A và B không còn dữ liệu gì cả |

Vd: Sử dụng IC giải mã và các cổng logic cơ bản sẵn có, ghép nối vi xử lý 8085 với bộ nhớ **ngoài dung lượng 10KB**, bao gồm:

**- Bắt đầu từ địa chỉ nhớ 0x8000** (8000h)

**- Từ các IC SRAM dung lượng 8 KB và 2 KB**

(8Kb từ các IC 4K => suy ra là dùng 2 IC 4K )

Giải:

B1:

IC SRAM:

IC1: 8KB = 8. 1024 = x =

=> Cần sử dụng 13 đường địa chỉ từ đến

IC2: 2KB= 2.1024 = 2x =

=> Cần sử dụng 11 đường địa chỉ từ đến

Địa chỉ bắt đầu: 0x8000 (hexa) = 1000 0000 0000 0000 (16-bit địa chỉ)

IC1: 8 KB (sử dụng 13 đường bit địa chỉ) chữ A là viết tắt của address

<= (bắt đầu từ bên phải )

Địa chỉ bắt đầu: 100|0 0000 0000 0000

Địa chỉ kết thúc: 100|1 1111 1111 1111

IC2: 2KB (sử dụng 11 đường địa chỉ) từ đến

(hiểu là IC 2 nối tiếp với IC 1 => địa chỉ kết thúc của IC1 sẽ cộng thêm 1 đơn vị => địa chỉ bắt đầu của IC2)

Có địa chỉ kết thúc của IC1 là: 100|1 1111 1111 1111 (tất cả các số đằng sau gạch | chuyển về bit 0: => 100|0 0000 0000 0000

100|~~0 0000 0000 0000~~ + 1 = 1010 0000 0000 0000

<=

Địa chỉ bắt đầu: 1010 0|000 0000 0000

Địa chỉ kết thúc: 1010 0|111 1111 1111

Bước cuối: vẽ sơ đồ kết nối dùng với các cổng logic cơ bản: and, or, not, xor,…

Vẽ vi xử lý 8085 kết nối với IC1 thì dùng địa chỉ kết thúc của IC1 (chỉ sử dụng các bit bên trái dấu gạch |)

Vẽ vi xử lý 8085 kết nối với IC2 thì dùng địa chỉ kết thúc của IC2 (chỉ sử dụng các bit bên trái dấu gạch |)

Đấu nối các chân địa chỉ sao cho tất cả đi qua cổng logic sẽ bằng mức 0

Câu 2: IC ngoài 8KB bằng các IC SRAM 4KB => phải dùng 2 con IC SRAM 4KB thì mới đủ 8KB

Với địa chỉ bắt đầu: 0x0000

IC SRAM:

IC1: 4KB = 4. 1024 = x =

=> Cần sử dụng 12 đường địa chỉ từ đến

IC2: 4KB= 4.1024 = x =

=> Cần sử dụng 12 đường địa chỉ từ đến

Địa chỉ bắt đầu: 0x0000 (hexa) = 0000 0000 0000 0000 (16-bit địa chỉ)

IC1: 4 KB (sử dụng 12 đường bit địa chỉ) chữ A là viết tắt của address

<= (bắt đầu từ bên phải)

Địa chỉ bắt đầu: 0000 |0000 0000 0000

Địa chỉ kết thúc: 0000 |1111 1111 1111

IC2: 4KB (sử dụng 12 đường địa chỉ) từ đến

(hiểu là IC 2 nối tiếp với IC 1 => địa chỉ kết thúc của IC1 sẽ cộng thêm 1 đơn vị => địa chỉ bắt đầu của IC2)

Có địa chỉ kết thúc của IC1 là: 0000 |1111 1111 1111 (tất cả các số đằng sau gạch | chuyển về bit 0: => 0000 |0000 0000 0000

0000| ~~0000 0000 0000~~ + 1 = 0001 0000 0000 0000

<=

Địa chỉ bắt đầu: 0001 |0000 0000 0000

Địa chỉ kết thúc: 0001 |1111 1111 1111

VD: 8Kb từ 2 Kb => cần dùng 4 con IC RAM 2KB

Địa chỉ bắt đầu 7000h

IC: 2KB = 2.1024= 2^11

=> sử dụng 11 đường địa chỉ từ Ao đến A 10

Địa chỉ bắt đầu: 7000h= 0111 0000 0000 0000

IC1: 2KB

Bắt đầu: 0111 0|000 0000 0000

Kết thúc: 0111 0|111 1111 1111

IC2: 2KB

Bắt đầu: 0111 1|000 0000 0000

Kết thúc: 0111 1|111 1111 1111

IC3: 2KB

Bắt đầu: 1000 0|000 0000 0000

Kết thúc: 1000 0|111 1111 1111

IC4: 2KB

Bắt đầu: 1000 1|000 0000 0000

Kết thúc: 1000 1|111 1111 1111

Vd: 6KB từ các IC 4 KB, => dùng 2 IC 4 KB => vượt mất 2 KB

Với địa chỉ bắt đầu là địa chỉ cao nhất trong không gian bộ nhớ (F000h)

Dùng: IC1= 4 KB,

IC2 (4KB) = 2 KB,

IC1= 4KB = 2^12

IC2 (4KB) = 2 KB = 2^11 (coi như là dùng IC 2 KB)

Địa chỉ bắt đầu: F000h = 1111 0000 0000 0000