**Câu 1. Trình bày kiến trúc & nguyên lý hoạt động của máy tính theo Von Neumann. Nêu vai trò của bộ nhớ và CPU trong hệ.**  
Nêu cơ chế mà nhờ đó CPU có thể thực hiện lệnh tuần tự.  
  
#Kiến trúc máy tính theo Von newmann gồm các thành phần sau:  
  
1. Bộ nhớ: Bộ nhớ máy tính là hệ thống con dùng lưu trữ các chỉ thị của chương trình và dữ liệu sẽ được thực hiện bởi máy tính. Thường gọi RAM, ROM. Trong kiến trúc Von Neumann chương trình và dữ liệu có chung không gian nhớ.  
2. Hệ thống vào/ra (I/O): Đây là hệ thống con cho phép máy tính tương tác với các thiết bị khác và liên lạc với thế giới bên ngoài.  
3. Hệ thống điều khiển trung tâm (CPU): Gồm có tập các thanh ghi (register) lưu trữ tạm thời thông tin, đơn vị số học và logic ALU thao tác trên các thông tin này, đơn vị giải mã lệnh và điều khiển xác định thao tác cần thực hiện và các hoạt động cần thiết để thực hiện thao tác, thanh ghi lệnh IR lưu mã nhị phân của lệnh cần thực thi, bộ đếm chương trình PC lưu địa chỉ của lệnh cần ghi.  
  
#Nguyên lý hoạt động:  
Dữ liệu (data) cùng với lệnh (instruction) được dùng để xử lí dữ liệu đó có thể được lưu giữ trên cùng một vùng nhớ của máy tính. Khi khối điều khiển Von Neumann bắt đầu gọi một lệnh để xử lí, nó gọi tới bộ đếm chương trình (program counter) để trỏ tới địa chỉ của lệnh trong bộ nhớ, rồi thì lệnh này sẽ được nạp và thực thi bởi vi xử lí. Địa chỉ của dữ liệu cần xử lí được chứa trong chính lệnh cần thực thi. Đang lúc tìm nạp và thực thị lệnh, bộ đếm chương trình sẽ tăng lên để trỏ tới lệnh tiếp theo cần thực thi. Quá trình này là tuần tự, nghĩa là các lệnh được thực thi một cách tuần tự, tại mỗi thời điểm chỉ có một lệnh được thực thi.  
  
#Vai trò của CPU và bộ nhớ trong hệ:  
- CPU thực hiện 2 thao tác: tìm nạp lệnh và thực thi lệnh.  
- Bộ nhớ lưu chương trình và dữ liệu.  
  
#Cơ chế thực hiện lệnh tuần tự: Đang lúc tìm nạp và thực thị lệnh, bộ đếm chương trình sẽ tăng lên để trỏ tới lệnh tiếp theo cần thực thi. Quá trình này là tuần tự, nghĩa là các lệnh được thực thi một cách tuần tự.  
  
**Câu 2. Nói rõ vì sao CPU có thể truy nhập theo địa chỉ. Nêu nguyên lý mà nhờ đó CPU có thể thực hiện lệnh rẽ nhánh có điều kiện.**  
# CPU có thể truy nhập theo địa chỉ vì dữ liệu, chương trình và các IO đều chia sẻ chung 1 không gian địa chỉ và được đánh địa chỉ cụ thể. CPU có thể truy nhập tất cả các thành phần này qua bus địa chỉ.  
  
#Khi gặp câu lệnh rẽ nhánh có điều kiện, vi xử lý sẽ kiểm tra điều kiện. Nếu điều kiện không được thỏa mãn, PC tăng lên 1 để thực hiện lệnh tiếp theo. Nếu điều kiện được thỏa mãn,chương trình sẽ nhảy đến nhãn được chỉ định. Lệnh rẽ nhánh có điều kiện xác định địa chỉ đích bằng kiểu định địa chỉ tương đối. Tức là địa chỉ đích trừ đi địa chỉ hiện tại được bao nhiêu cộng vào PC. Giới hạn khoảng cách nhảy từ -128 đến +127 byte kể từ lệnh sau lệnh nhảy có điều kiện.

**Câu 3. Nêu rõ vì sao có thể nối chung bus số liệu trong hệ. Bạn hiểu thế nào khi CPU truy cập vổng vào ra như 1 ô nhớ, hay truy cập ô nhớ nhờ cổng vào ra.**  
#Các vùng nhớ dữ liệu và chương trình có cùng không gian địa chỉ nên có thể nối bus chung.  
#Các port của vi điều khiển được đánh địa chỉ và lưu trong thanh ghi SFR. Nhờ vậy các cổng được truy nhập thông qua địa chỉ như 1 vị trí trong bộ nhớ. Khi vi xử lý có kết nối bộ nhớ ngoài, địa chỉ bộ nhớ sẽ được truyền dẫn giữa CPU và các mạch giải mã thông qua các port. Sau khi xác định ô nhớ, dữ liệu cũng được truyền dẫn giữa CPU và ô nhớ thông qua các port. Vậy là CPU truy nhập ô nhớ nhờ cổng vào ra.  
  
**Câu 4. Bằng cách nào CPU có thể truy nhập tới các phần tử khác nhau thông qua địa chỉ khác nhau.**  
#Các địa chỉ được biên dịch sang mã máy, được truyền dẫn thông qua bus địa chỉ và nhờ 1 hệ thống mạch logic tổ hợp làm hệ thống giải mã địa chỉ mà xác định đường đến phần tử tương ứng.

**Câu 5. Mô tả chi tiết hoạt động thực hiện 1 lệnh của CPU.**  
Minh họa trường hợp truy nhập trực tiếp  
MOV R0,50H  
# Việc thực hiện 1 lệnh của CPU như sau:  
- Nội dung của PC được đặt lên bus địa chỉ.  
- Tín hiệu điều khiển read được xác lập.  
- Opcode của lệnh được đọc từ RAM và đưa lên bus dữ liệu.  
- Opcode được chốt vào thanh ghi lệnh bên trong CPU.  
- PC được tăng để chuẩn bị tìm nạp lệnh kế tiếp từ bộ nhớ.  
- Khối giải mã lệnh giải mã opcode và tạo tín hiệu điều khiển. ALU nhận tín hiệu và thực thi lệnh.  
#Minh họa trường hợp truy nhập trực tiếp:  
- Địa chỉ nơi lưu mã opcode của lệnh MOV được PC đặt lên bus địa chỉ.  
- Xác lập tín hiệu read.  
- Opcode của lệnh được đọc từ RAM và đưa lên bus dữ liệu: 1010 1000. Lệnh này có mã opcode 10101rrr, rrr từ 000 đến 111 mã hóa 7 thanh ghi từ R0 đến R7.  
- Thanh ghi lệnh được chốt nội dung 1010 1000  
- PC tăng lên  
- Khối giải mã lệnh và diều khiển giải mã opcode này, tạo ra các tín hiệu điều khiển, khối ALU thực hiện các thao tác đã định. Quá trình này bao gồm việc đưa 50H lên bus địa chỉ, đọc nội dung ô nhớ từ Ram và đưa lên bus dữ liệu, nhận dữ liệu đọc được và chuyển vào R0, thông báo thực hiện lệnh tiếp theo.  
  
**Câu 6: Minh họa trường hợp đọc toán hạng ở chế độ gián tiếp qua thanh ghi.**  
ADD A,@R0  
#Minh họa:  
- Địa chỉ nơi lưu mã opcode của lệnh MOV được PC đặt lên bus địa chỉ.  
- Xác lập tín hiệu read.  
- Opcode của lệnh được đọc từ RAM và đưa lên bus dữ liệu: 0010 0110. Lệnh này có mã opcode 0010 011i, I có giá trị 0 hoặc 1 mã hóa 2 thanh ghi R0 và R1.  
- Thanh ghi lệnh được chốt nội dung 0010 0110  
- PC tăng lên  
- Khối giải mã lệnh và diều khiển giải mã opcode này, tạo ra các tín hiệu điều khiển, khối ALU thực hiện các thao tác đã định. Quá trình này bao gồm việc đọc dữ liệu ở R0, đặt dữ liệu này lên bus địa chỉ, đọc nội dung ô nhớ có địa chỉ lưu ở R0 rồi đưa lên bus dữ liệu, nhận dữ liệu đọc được và chuyển vào thanh ghi A, thông báo thực hiện lệnh tiếp theo.

**Câu 7. Mô tả chi tiết trường hợp đọc toán hạng ở chế độ tức thì.**  
MOV A,#50H  
#Mô tả:  
- Địa chỉ nơi lưu mã opcode của lệnh MOV được PC đặt lên bus địa chỉ.  
- Xác lập tín hiệu read.  
- Opcode của lệnh được đọc từ RAM và đưa lên bus dữ liệu: 0111 0100. Do ở chế độ định địa chỉ tức thì, giá trị 50H được lưu ở ô nhớ ngay sau ô nhớ lưu opcode của lệnh và được đưa lên bus sau khi opcode của lệnh được chốt. Bộ nhớ Ram chỉ được đọc 1 lần.  
- Thanh ghi lệnh được chốt nội dung 0111 0100  
- PC tăng lên  
- Khối giải mã lệnh và diều khiển giải mã opcode này, tạo ra các tín hiệu điều khiển, khối ALU thực hiện các thao tác đã định. Quá trình này bao gồm các việc cần thiết để chốt giá trị 50H vào thanh ghi A và thông báo thực hiện lệnh tiếp theo.   
  
**Câu 8. Minh họa trường hợp đọc toán hạng ở chế độ thanh ghi.**  
SUBB A,R7  
#Minh họa:  
- Địa chỉ nơi lưu mã opcode của lệnh SUBB được PC đặt lên bus địa chỉ.  
- Xác lập tín hiệu read.  
- Opcode của lệnh được đọc từ RAM và đưa lên bus dữ liệu: 1001 1111. Lệnh này có mã opcode là 10011rrr, với rrr mã hóa từ 000 đến 111 tương ứng với các thanh ghi R0 đến R7  
- Thanh ghi lệnh được chốt nội dung 1001 1111  
- PC tăng lên  
- Khối giải mã lệnh và diều khiển giải mã opcode này, tạo ra các tín hiệu điều khiển, khối ALU thực hiện các thao tác đã định. Quá trình này bao gồm các việc cần thiết để trừ nội dung thanh chứa A bởi nội dung của thanh ghi R7 và cờ nhớ rồi lưu kết quả vào A và thông báo thực hiện lệnh tiếp theo.

**Câu 9. Nêu cơ chế ngắt trong vi xử lý, giải pháp cất giữ và khôi phục trạng thái. Trình bày cách xác định nguồn báo ngắt bằng phần mềm.**  
#Cơ chế ngắt, giải pháp cất giữ và khôi phục trạng thái:  
Khi chương trình chính đang thực hiện mà có 1 ngắt xuất hiện,CPU sẽ kiểm tra xem ngắt có được phép hay không, nếu được phép, việc thực thi chương trình chính tạm dừng và CPU thực hiện việc rẽ nhánh đến chương trình phục vụ ngắt ISR. CPU thực thi ISR để thực hiện 1 công việc và kết thúc việc thực thi này khi gặp lệnh “ quay về từ 1 trình phục vụ ngắt”, chương trình chính được tiếp tục tại nơi bị tạm dừng.  
Để cất giữ và khôi phục trạng thái, khi xảy ra ngắt, giá trị của PC được lưu vào ngăn xếp và dược nạp lại vào PC sau khi hoàn thành chương trình ngắt.  
# Xác định nguồn báo ngắt bằng phần mềm:  
Để xác định nguồn báo ngắt bằng phần mềm, trong chương trình phục vụ ngắt cần có câu lệnh kiểm tra trạng thái của các vector ngắt, trạng thái của thanh ghi cho phép ngắt và thanh ghi mức ưu tiên của các ngắt. Nếu các giá trị đều trùng nhau ta xác định dựa vào chuỗi vòng cố định, ngắt nào đứng trước là ngắt được phục vụ: ngắt ngoài 0, ngắt do timer 0, ngắt ngoài 1, ngắt do timer 1, ngắt do truyền tin nối tiếp, ngắt do timer2.  
  
**Câu 10. Nêu ưu, khuyết điểm của cơ chế ngắt trong VXL. Trình bày cách xác định nguồn báo ngắt bằng phần cứng.**  
# Cơ chế ngắt có ưu điểm là giúp chương trình không phải liên tục kiểm tra cờ ngắt nên có thể thoải mái thực hiện các công việc khác khi không có ngắt xảy ra.  
Nhược điểm là quá trình lưu trạng thái và khôi phục trạng thái dễ gây ra lỗi, đặc biệt là nếu trong chương trình phục vụ ngắt có tác động đến ngăn xếp. Chương trình có sử dụng ngắt có hoạt động phức tạp gây khó khăn , nhầm lẫn cho người lập trình.  
#Cách xác định nguồn báo ngắt bằng phần cứng: ??? Bạn nào biết trả lời hộ mình câu này

**Câu 11. Vai trò và hoạt động của ngăn xếp. Liệu có tồn tại hệ vi xử lý hoạt động mà không cần sự tồn tại của ngăn xếp không? Giải thích?**  
#Ngăn xếp có tác dụng lưu giữ trạng thái của PC khi chương trình xuất hiện ngắt, hoặc các câu lệnh gọi hàm. Ngăn xếp cũng có thể dùng để lưu 1 số dữ liệu tạm thời.  
Ngăn xếp hoạt động theo cơ chế fisrt in- last out, dữ liệu nào được lưu và trước thì phải được lấy ra sau. Ngăn xếp sử dụng 1 con trỏ ngăn xếp SP trỏ đến địa chỉ lưu giữ liệu. Khi lưu giữ liệu, trước khi thực hiện mỗi lệnh ghi PUSH, con trỏ ngăn xếp tự động tăng 1 đến ngăn nhớ tiếp theo. Khi lấy dữ liệu, trước khi thực hiện mỗi lệnh POP, SP tự động giảm đi 1. Con trỏ SP mặc định là trỏ đến 07H, tức là lệnh cất đầu tiên sẽ cất vào ô nhớ 08H.  
#Theo kiến trúc Von Neumann, VXL tại 1 thời điểm chỉ thực hiện 1 công việc. Vậy để có thể thực hiện được lệnh gọi các chương trình con hoặc chương trình ngắt thì bắt buộc phải có ngăn xếp hoặc 1 cơ chế nào đó tương tự ngăn xếp để lưu trạng thái làm việc .  
  
**Câu 12. Vẽ lược đồ hoạt động của VXL kể từ lúc RESET. Vector reset và vector ngắt có thể trùng nhau không? Giải thích. Nêu sự cần thiết của ngắt không che.**  
#Lược đồ hoạt động của VXL kể từ lúc reset???  
#Vector reset và vetor ngắt có thể trùng nhau. Vì thực ra reset cũng coi là 1 hoạt động ngắt. Khi chân reset ở mức cao trong tối thiểu 2 chu kì rồi chuyển về mức thấp, VXL sẽ reset và PC nạp giá trị của vector reset là 0000H.  
#Ngắt không che là cần thiết. Ngắt không che hữu dụng trong các trường hợp VXL gặp sự cố. 8051 có 1 ngắt không che được là reset.

**Câu 13. Kiến trúc phần cứng của hệ VXL. Vai trò của mạch giải mã địa chỉ. Bằng cách nào CPU có thể trao đổi với các thiết bị ngoại vi.**  
#Kiến trúc phần cứng giống y chang kiến trúc Von Neuman. Chỉ khác ở chỗ chưa xác định không gian bộ nhớ dữ liệu và chương trình chung hay riêng.  
#Khi giao tiếp với các thiết bị ngoại vi, cần có mạch giải mã địa chỉ tạo ra tín hiệu chip select để chọn thiết bị ngoại vi nào sẽ giao tiếp với CPU.  
#Mỗi thiết bị ngoại vi được đánh địa chỉ theo kiểu ánh xạ bộ nhớ. Vì vậy CPU có thể thao tác với ngoại vi như thao tác với 1 ngăn nhớ.  
  
**Câu 14. Trình bày phương thức điều khiển vào ra bằng chương trình, điều khiển ở đây là điều khiển gì? Nêu ưu khuyết điểm của phương thức này.**  
#Để điều khiển vào ra, ta có thể sử dụng vòng lặp trong chương trình. Trong chương trình ta dùng các vòng lặp để CPU dành ra 1 khoảng thời gian quét dữ liệu ở cồng vào hoặc các cờ ngắt để xác định xem có ngoại vi nào cần giao tiếp không. Nếu có thì sẽ gọi các chương trình con để thao tác.  
#Phương pháp này có ưu điểm là cấu trúc chương trình đơn giản. Nhược điểm là CPU sẽ tốn thời gian liên tục kiểm tra các ngoại vi kể cả khi không có ngoại vi nào yêu cầu giao tiếp.

**Câu 15. Trình bày phương thức điều khiển vào ra bằng ngắt. Nêu ưu khuyết điểm của phương thức này, nêu điểm giống và khác so với điều khiển bằng chương trình.**  
#Khi thiết bị vào ra có nhu cầu trao đổi thông tin với CPU, thiết bị có thể gửi tín hiệu yêu cầu giao tiếp vào chân ngắt ngoài của VXL. Khi đó CPU sẽ kiểm tra ngắt xem có được phép hay không. Nếu ngắt được phép, CPU sẽ dừng chương trình đang thực hiện lại và nhảy đến chương trình ngắt. Ta có thể viết các lệnh điều khiển vào ra trong chương trình ngắt.  
#Ưu điểm của chương trình là tiết kiệm thời gian, CPU có thể thực hiện các công việc khác khi không điều khiển vào ra. Nhược điểm là việc lập trình phức tạp hơn. Cách này khác điều khiển bằng chương trình ở chỗ các lệnh điều khiển vào ra nằm ở chương trình ngắt, CPU sẽ gọi như gọi chương trình con, còn cách điều khiển vào ra bằng chương trình các lệnh điều khiển vào ra nằm trong chương trình chính. Điểm giống nhau là các lệnh điều khiển đều như nhau.  
  
**Câu 16. Trình bày điều khiển vào ra bằng cách truy nhập trực tiếp bộ nhớ. Nêu ưu khuyết điểm của phương pháp này. Điểm khác chính so với 2 phương pháp kia là gì.**  
  
#Cách truy nhập trực tiếp bộ nhớ cần thêm 1 thiết bị phụ trợ điều khiển việc truy nhập dữ liệu DMAC (Direct Memory Access Controller). . Cấu tạo của DMAC bao gồm:   
- Thanh ghi dữ liệu chứa giữ liệu trao đổi.  
- Thanh ghi địa chỉ chứa địa chỉ ngăn nhớ.  
- Bộ đếm dữ liệu chứa số bite dữ liệu cần trao đổi.  
- Khối logic điều khiển điều khiển hoạt động của DMAC.  
Khi IO cần trao đổi với bộ nhớ, nó gửi yêu cầu đến DMAC, DMAC sẽ yêu cầu VXL dừng việc điều khiển bus dữ liệu và bus địa chỉ, việc này được trao cho DMAC. Địa chỉ thiết bị vào ra và bộ nhớ được cung cấp bởi CPU. Địa chỉ đầu của mảng dữ liệu được lưu vào thanh ghi địa chỉ. Số byte dữ liệu cần trao đổi lưu vào bộ đếm dữ liệu. Dữ liệu cần truyền nạp vào thanh ghi dữ liệu. Sau khi truyền xong 1 byte dữ liệu, bộ đếm giảm xuống 1, địa chỉ ngăn nhớ tăng lên 1. Khi bộ đếm bằng 0 DMAC sẽ thông báo truyền xong dữ liệu và trả quyền điều khiển về CPU.  
# Phương pháp này ưu điểm là truyền dữ liệu có kích thước lớn với tốc độ nhanh. Nhược điểm là thiết kế phức tạp. Điểm khác biệt chính của phương pháp này là CPU không tham gia vào quá trình truyền dữ liệu.

**Câu 17. Trình bày nguyên lý của cổng vào không đối thoại, có đối thoại.**  
  
# Mạch cổng vào không đối thoại gồm ngoại vi, bộ đệm 3 trạng thái, mạch giải mã địa chỉ, CPU. Dữ liệu phải luôn thường trực ở ngoại vi. Khi cần đọc dữ liệu, CPU phát địa chỉ vào mạch giải mã địa chỉ. Mạch giải mã sẽ xác định CPU muốn lấy dữ liệu từ ngoại vi nào mà phát tín hiệu chip select tương ứng. Tín hiệu CS được OR với tín hiệu điều khiển đọc RD và đi vào chân OE của bộ đệm 3 trạng thái làm bộ đệm cho dữ liệu đi qua. Dữ liệu từ ngoại vi qua bộ đệm vào CPU.  
  
Mạch cổng vào có đối thoại có thêm D flip-flop và mạch chốt dữ liệu. Khi ngoại vi cần gửi dữ liệu, ngoại vi phát dữ liệu lên bus và tín hiệu STB vào mạch chốt dữ liệu để chốt dữ liệu. Tín hiệu STB cũng được phát vào chân EN của D flip-flop. Đầu vào D của flip-flop treo lên 1. Khi có tin hiệu STB,đầu ra flip-flop phát tín hiệu IBF vào chân INT của CPU. CPU gọi chương trình ngắt nhận dữ liệu tương ứng. CPU phát tín địa chỉ ngoại vi vào mạch giải mã. Mạch giải mã phát tín hiệu CS. Tín hiệu CS được OR với tín hiệu RD của CPU đi vào chân OE của bộ đệm 3 trạng thái. Bộ đệm 3 trạng thái thông cho dữ liệu đi từ mạch chốt dữ liệu vào CPU. Tín hiệu đi vào chân OE đồng thời đi vào chân Clear của D flip-flop làm tín hiệu IBF trở về 0. Tín IBF được đưa về ngoại vi báo xử lý xong dữ liệu.

**Câu 18. Trình bày cổng ra có đối thoại, không đối thoại.**  
# Trường hợp ra không đối thoại cơ bản giống với vào không đối thoại, chỉ có 2 thay đổi nhỏ. Dữ liệu đi từ CPU qua bộ chốt có đầu ra 3 trạng thái tới ngoại vi. Tín hiệu RD thay bằng tín hiệu WR và được OR với tín hiệu CS đi vào bộ chốt.  
# Ở cổng ra song song có đối thoại, dữ liệu đi qua bộ chốt, bộ đệm tới ngoại vi. Khi có gửi dữ liệu, VXL nhảy tới chương trình ngắt điều khiển việc gửi dữ liệu. VXL phát địa chỉ tới mạch giải mã địa chỉ, tín hiệu điều khiển WR, và dữ liệu tới bộ chốt. Bộ giải mã phát tín hiệu CS được OR với tín hiệu WR tới bộ chốt để bộ chốt chốt dữ liệu. Tín hiệu này đồng thời được gửi tới chân CLK của D flip-flop. D flip-flop có đầu vào treo lên 1, khi có xung CLK đầu ra sẽ phát tin hiệu OBF tới chân INT của VXL báo hiệu đã chốt dữ liệu để VXL quay về chương trình chính làm việc khác. Tín hiệu OBF cũng được gửi tới ngoại vi để báo có tín hiệu gửi từ VXL. Ngoại vi sẽ gửi tín hiệu ACK tới bộ đệm để cho dữ liệu đi từ bộ chốt qua được bộ đệm tới ngoại vi. Tín hiệu ACK đồng thời gửi tới chân CLR của flip-flop để xóa tín hiệu OBF xuống mức thấp. Tín hiệu OBF được gửi tới chân INT của VXL để VXL nhảy vào chương trình ngắt truyền dữ liệu tiếp theo.

**Câu 19. Trình bày nguyên lý hoạt động của cổng ra read back. Tác dụng của cổng này. Trình bày nguyên lý hoạt động của cổng vào ra có điều khiển hướng: theo nhóm 8 bit, theo từng bit.**  
  
#Cổng ra read back có cấu tạo cơ bản giống với cổng ra thông thường, tuy nhiên có sự khác biệt là bus dữ liệu có thêm 1 nhánh dẫn dữ liệu qua 1 bộ đệm và quay ngược lại cổng vừa xuất ra. Sau khi gửi dữ liệu, VXL sẽ phát tín hiệu RD. Tín hiệu này được OR với tín hiệu CS từ bộ giải mã địa chỉ tới chân OE của bộ đệm. Bộ đệm thông và VXL đọc lại dữ liệu được xuất ra.  
#Cổng ra read back giúp VXL kiểm tra được dữ liệu vừa xuất ra có chính xác hay không.  
#Đối với cổng ra có điều khiển theo hướng 8 bit, các mạch ngoài bắt buộc phải ghép nối theo kênh bus, dữ liệu truyền đi bắt buộc phải theo cả bite. Trong khi điều khiển theo từng bit thì mạch ngoài có thể ghép nối khá tùy tiện vào các chân Pin và dữ liệu có thể truyền theo từng bit.

**Câu 20. Vào ra nối tiếp: định nghĩa, chế độ, nguyên lý, mục đích, phương thức, nhịp truyền. Dùng cổng song song có thể vào ra nối tiếp được không? Giải thích.**  
  
#Vào ra dữ liệu nối tiếp là phương pháp và ra dữ liệu theo từng bit. Nguyên lý chung là sử dụng thanh ghi dịch 8 bit biến dữ liệu song song thành nối tiếp ( bên phát) và ngược lại ( bên thu ). Có 3 chế độ:  
-Đơn công: bên phát chỉ phát, bên thu chỉ thu.  
-Bán song công: cả 2 bên đều có thể thu hoặc phát. Tại 1 thời điểm chỉ thu hoặc chỉ phát.  
-Song công hoàn toàn: cả 2 bên đều có thể thu phát đồng thời.  
Mục đích của phương thức truyền nối tiếp là tiết kiệm nguyên vật liệu, có thể truyền tín hiệu đi xa. Phương thức là truyền lần lượt 8 bit với 1 bít mở đầu và 1 bit kết thúc tạo thành 1 khung, có thể có thêm bit chẵn lẻ. Nhịp truyền là số bit truyền được trong 1s.  
#Trong cổng chuyên dùng để vào ra nối tiếp có thanh ghi dịch 8 bit, cổng thông thường không có nên không dùng cổng song song truyền tin nối tiếp được.

**Câu 21. Trình bày về cổng vào ra nối tiếp trong hệ MCS-51. Hãy trình bày rõ cách sử dụng bit Parity trong trường hợp này.**  
  
#Port nối tiếp của MCS gồm chân TxD ( P3.1 ) để phát dữ liệu và RxD (P3.0) để thu dữ liệu. Port hoạt động theo chế độ song công. Ngoài ra, port cho phép 1 kí tự được nhận và lưu trong bộ dệm trong khi kí tự tiếp theo được nhận vào. Nếu CPU đọc kí tự thứ nhất trước khi ký tự thứ 2 dược nhận đầy đủ, dữ liệu sẽ không bị mất.   
Phần mềm sử dụng 2 thanh ghi chức năng đặc biệt SBUF và SCON để truy xuất port nối tiếp.  
SBUF gồm 2 bộ đệm, 1 cho phát và 1 cho thu. Việc ghi lên SBUF sẽ nạp dữ liệu để phát, việc đọc SBUF sẽ lấy dữ liệu nhận được.  
Thanh ghi SCON được định địa chỉ bit, thanh ghi chứa các bit trạng thái và bit điều khiển. Các bit điều khiển sẽ thiết lập chế độ hoạt động cho các port nối tiếp còn các bit trạng thái sẽ chỉ ra sự kết thúc việc thu hoặc phát. Các bit trạng thái được kiểm tra bằng phần mềm hoặc được lập trình để tạo ra ngắt.  
# Bit kiểm tra chẵn lẻ thường được sử dụng ở chế độ 2 và 3. Bit này là bit có kí hiệu P nằm trong thanh ghi đặc biệt PSW. Bit này cần được nạp vào bit có kí hiệu TB8 của thanh ghi SCON bằng phần mềm. Cụ thể nếu kiểm tra chẵn ta dùng đoạn lệnh sau: MOV C,P  
MOV TB8,C  
MOV SBUF,A  
Nếu kiểm tra chẵn ta dùng đoạn lệnh sau:  
MOV C,P  
CPL C  
MOV TB8,C  
MOV SBUF,A

**Câu 22. Vào ra nối tiếp dị bộ: định nghĩa, mục đích, phương pháp đồng bộ. Nêu mặt hạn chế của chuẩn RS232C.**  
#Vào ra nối tiếp dị bộ là phương pháp truyền tin khi mà xung clock riêng rẽ ở bộ phận thu và phát. Mục đích chính là để khi phát tin không phải phát theo xung clock đồng bộ khá phức tạp. Phương pháp đồng bộ là đồng bộ theo từng bite bằng cách thêm vào các bit khởi đầu và bit kết thúc.  
#Chuẩn RS232C có hạn chế là mức logic 0 là 3đến 12V, mức 1 là -3 đến -12V, mức điện áp này không phù hợp với các chip TTL phổ biến, vì vậy cần có 1 mạch trung gian MAX 232

**Câu 23. Mạch thời gian trong VXL: công dụng, cấu trúc, hoạt động và ghép nối với hệ VXL. Nếu dùng mạch thời gian để đếm sự kiện thì ta phải làm gì? Nêu sự khác biệt giữa mạch thời gian và mạch đếm.**  
# Mạch thời gian trong VXL có thể dùng để đếm sự kiện, đo thời gian hoặc tạo tốc độ baud cho truyền tin nối tiếp. Cấu tạo của mạch đếm thời gian là 1 chuỗi các flip-flop nối tiếp với mỗi flip flop là 1 mạch chia 2 tần số clock, flip flop cuối có ngõ ra là cờ tràn. Trạng thái của mỗi flip flop là 1 bit trong số đếm của mạch thời gian.  
  
#Nếu dùng mạch thời gian để đếm sự kiện, ta đặt nguồn tạo xung clock tạo xung dựa trên việc xảy ra 1 sự kiện.  
  
#Trong mạch thời gian, xung clock do 1 mạch dao động cung cấp, bộ đếm hoạt động đều đặn theo chu kì dao động của mạch. Trong mạch đếm, mỗi khi có 1 sự kiện xảy ra thì bộ đếm mới hoạt động.

**Câu 24. Mạch thời gian trong bộ MCS 51 của intel: mô tả các thanh ghi liên quan. Các chế độ hoạt động và ứng dụng của các chế độ đó. Sử dụng ngắt với mạch thời gian như thế nào?**  
  
# Các thanh ghi liên quan:  
-TCON điều khiển hoạt động và trạng thái của các bộ định thời. TF0,TF1 báo các bộ định thời tràn. TR0,TR1 cho phép dừng hoặc hoạt động các bộ định thời. IE0,IE1 thông báo có ngắt ngoài. IT1,IT0 chọn kiểu ngắt ngoài là theo cạnh hay mức.  
-Thanh ghi chế độ định thời TMOD chọn chế độ hoạt động cho định thời. GATE điều khiển cổng: khi set lên 1, bộ định thời hoạt độg khi chân INT ở mức cao. C/T chọn chức năng đếm hoặc định thời. M0,M1 chọn chế độ định thời.  
-TL0,TH1 lần lượt là các thanh ghi lưu bite thấp và bite cao của bộ định thời 0. TL1,TH1 lần lượt là các thanh ghi lưu bite thấp và bite cao của bộ định thời 1.  
  
#Chế độ 0 là chế độ định thời 13 bit. Byte cao THx và 5 bit thấp của byte thấp TLx được ghép với nhau tạo thành bộ định thời 13 bit. Chế độ này tương thích với bộ vi điều khiển cũ 8048, hiện hiếm dùng.  
Chế độ 1 là chế độ định thời 16 bit. Chế độ này đếm được từ 0000h đếm FFFFh, tức là từ 0 đến 65536. Chế độ này ứng dụng cho những bộ đếm có thời gian đếm dài. Đặc biệt hữu hiệu khi kết hợp với vòng lặp.  
Chế độ 2 là chế độ 8 bit tự nạp lại. TLx đếm từ giá trị được nạp đến FFh sẽ tràn và cờ tràn được set lên 1. Sau đó giá trị của THx được nạp lại vào TLx và tiếp tục đếm. Chế độ này thuận lợi khi cần đếm thời gian tuần hoàn và chính xác cao.  
Chế độ 3 là chế độ chia sẻ giữa timer 0 và 1. Khi hoạt động ở chế độ 3, TH0 và TL0 đếm thời gian riêng rẽ. Khi TH0 tràn, TF1 được set. Khi TL0 tràn TF0 được set. Bộ định thời 1 có thể hoạt động nhưng không ảnh hưởng đến cờ tràn TF1. Khi dùng chế độ này timer 1 có thể được dùng vào những việc không cần đến cờ tràn.  
  
#Khi xảy ra tràn bộ định thời, VXL sẽ thực hiện việc kiểm tra xem ngắt có hợp lệ hay không. Nếu hợp lệ chương trình sẽ nhảy đến chương trình phục vụ ngắt và tự động xóa cờ tràn.