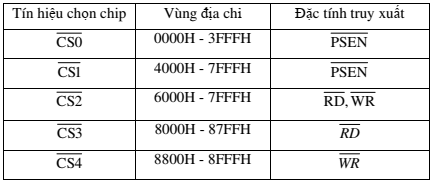
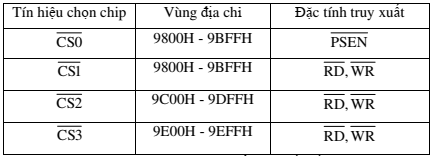
**BÀI TẬP VI ĐIỀU KHIỂN 8051**

1. Sử dụng 1 vi mạch 74138 và các cổng cần thiết để thiết kế mạch giải mã địa chỉ tạo ra các tín hiệu chọn chip tương ứng các vùng địa chỉ sau:



2. Sử dụng 1 vi mạch 74138 và các cổng cần thiết để thiết kế mạch giải mã địa chỉ tạo ra các tín  
hiệu chọn chip tương ứng các vùng địa chỉ sau:



1. Chỉ dùng một vi mạch 74138 (không dùng thêm cổng), thiết kế mạch giải mã địa chỉ tạo ra một  
   tín hiệu chọn chip /CS tương ứng tầm địa chỉ F000H-F3FFH.

**SỬ DỤNG TẬP LỆNH**

***TẤT CẢ CHUYỂN QUA NGÔN NGỮ C***

Truy xuất RAM nội2.1 Viết CT ghi 40H vào ô nhớ 30H của RAM nội theo 2 cách (định địa chỉ ô nhớ trực tiếp và định  
địa chỉ ô nhớ gián tiếp).  
Cách 1: Định địa chỉ trực tiếp  
ORG 0000H  
MOV 30H,#40H  
END  
Cách 2: Định địa chỉ gián tiếp  
ORG 0000H  
MOV R0,#30H  
MOV @R0,#40H  
END

2.2 Viết CT xóa ô nhớ 31H của RAM nội theo 2 cách (định địa chỉ ô nhớ trực tiếp và định địa chỉ ô  
nhớ gián tiếp).  
HD: Xóa là ghi giá trị 0.

2.3 Viết CT ghi nội dung thanh ghi A vào ô nhớ 32H của RAM nội theo 2 cách (định địa chỉ ô nhớ  
trực tiếp và định địa chỉ ô nhớ gián tiếp).  
CÁCH 1:  
ORG 0000H  
MOV 32H,A  
END  
CÁCH 2:  
ORG 0000H  
MOV R0,#32H  
MOV @R0,A  
END

2.4 Viết CT đọc ô nhớ 33H của RAM nội vào thanh ghi A theo 2 cách (định địa chỉ ô nhớ trực tiếp  
và định địa chỉ ô nhớ gián tiếp).  
CÁCH 1:  
ORG 0000H  
MOV A,33H  
END  
CÁCH 2:  
ORG 0000H  
MOV R0,#33H  
MOV A,@R0  
END

2.5 Viết CT chuyển dữ liệu ô nhớ 34H của RAM nội vào ô nhớ 35H của RAM nội theo 2 cách  
(định địa chỉ ô nhớ trực tiếp và định địa chỉ ô nhớ gián tiếp).  
CÁCH 1:  
ORG 0000H  
MOV 35H,34H  
END  
CÁCH 2:  
ORG 0000H  
MOV R0,#34H  
MOV A,@R0  
INC R0  
MOV @R0,A  
END  
Truy xuất RAM ngoài

2.6 Viết CT ghi 40H vào ô nhớ 0030H của RAM ngoài.

ORG 0000H  
MOV A,#40H  
MOV DPTR,#0030H  
MOVX @DPTR,A  
END  
2.7 Viết CT xóa ô nhớ 0031H của RAM ngoài.  
HD: Ghi giá trị 0.  
2.8 Viết CT đọc ô nhớ 0032H của RAM ngoài vào thanh ghi A.  
ORG 0000H  
MOV DPTR,#0032H  
MOVX A,@DPTR  
END  
2.8 Viết CT ghi nội dung thanh ghi A vào ô nhớ 0033H của RAM ngoài.  
ORG 0000H  
MOV DPTR,#0033H  
MOVX @DPTR,A  
END  
2.10 Viết CT chuyển dữ liệu ô nhớ 0034H của RAM ngoài vào ô nhớ 0035H của RAM ngoài.  
ORG 0000H  
MOV DPTR,#0034H  
MOVX A,@DPTR  
INC DPTR  
MOVX @DPTR,A  
END  
Truy xuất Port2.11 Viết CT xuất 0FH ra Port 1.  
ORG 0000H  
MOV P1,#0FH  
END  
2.12 Viết CT xuất F0H ra Port 2.  
2.13 Viết CT xuất nội dung thanh ghi A ra Port 1.  
ORG 0000H  
MOV P1,A  
END  
2.14 Viết CT nhập từ Port 1 vào thanh ghi A.  
ORG 0000H  
MOV A,P1  
END  
2.15 Viết CT nhập từ Port 1 và xuất ra Port 2

2.16 Viết CT xuất 1 (mức logic cao) ra chân P1.0  
ORG 0000H  
SETB P1.0  
END  
2.17 Viết CT xuất 0 (mức logic thấp) ra chân P1.1  
ORG 0000H  
CLR P1.1  
END  
Truy xuất RAM nội, RAM ngoài và Port2.18 Viết CT chuyển dữ liệu ô nhớ 40H (RAM nội) đến ô nhớ 2000H (RAM ngoài). Làm theo 2  
cách (định địa chỉ ô nhớ trực tiếp và định địa chỉ ô nhớ gián tiếp).  
CÁCH 1:  
ORG 0000H  
MOV A,40H  
MOV DPTR,#2000H  
MOVX @DPTR,A  
END  
CÁCH 2:  
ORG 0000H  
MOV R0,#40H  
MOV DPTR,#2000H  
MOV A,@R0  
MOVX @DPTR,A  
END  
2.19 Viết CT chuyển dữ liệu ô nhớ 2001H (RAM ngoài) vào ô nhớ 41H (RAM nội). Làm theo 2  
cách (định địa chỉ ô nhớ trực tiếp và định địa chỉ ô nhớ gián tiếp).  
2.20 Viết CT nhập từ Port 1 vào ô nhớ 42H (RAM nội). Làm theo 2 cách (định địa chỉ ô nhớ trực  
tiếp và định địa chỉ ô nhớ gián tiếp).  
CÁCH 1:  
ORG 0000H  
MOV 42H,P1  
END  
CÁCH 2:  
ORG 0000H  
MOV R0,#40H  
MOV @R0,P1  
END  
2.21 Viết CT nhập từ Port 1 vào ô nhớ 2002H (RAM ngoài).  
ORG 0000H  
MOV DPTR,#2002H

MOV A,P1  
MOVX @DPTR,A  
END  
2.22 Viết CT lấy ô nhớ 43H (RAM nội) xuất ra Port 1. Làm theo 2 cách (định địa chỉ ô nhớ trực  
tiếp và định địa chỉ ô nhớ gián tiếp).  
2.23 Viết CT lấy ô nhớ 2003H (RAM ngoài) xuất ra Port 1.  
Sử dụng vòng lặp2.24 Viết CT xóa 20 ô nhớ RAM nội có địa chỉ bắt đầu là 30H.  
ORG 0000H  
MOV R0,#30H  
MOV R2,#20  
laplai:  
MOV @R0,#0  
INC R0  
DJNZ R2,laplai  
END  
2.25 Viết CT xóa các ô nhớ RAM nội từ địa chỉ 20H đến 7FH.  
ORG 0000H  
MOV R0,#20H  
laplai:  
MOV @R0,#0  
INC R0  
CJNE R0,#80H,laplai  
END  
2.26 Viết CT xóa 250 ô nhớ RAM ngoài có địa chỉ bắt đầu là 4000H.  
ORG 0000H  
MOV DPTR,#4000H  
CLR A  
MOV R7,#250  
lap1: MOVX @DPTR,A  
INC DPTR  
DJNZ R7,lap1  
END  
2.27 Viết CT xóa 2500 ô nhớ RAM ngoài có địa chỉ bắt đầu là 4000H.  
ORG 0000H  
MOV DPTR,#4000H  
CLR A  
MOV R6,#10  
lap2: MOV R7,#250

lap1: MOVX @DPTR,A  
INC DPTR  
DJNZ R7,lap1  
DJNZ R6,lap2  
END  
2.28 Viết CT xóa toàn bộ RAM ngoài có dung lượng 8KB, biết rằng địa chỉ đầu là 2000H.  
HD: 8KB = 8192Byte  
2.29 Viết CT chuyển một chuỗi dữ liệu gồm 10 byte trong RAM nội có địa chỉ đầu là 30H đến vùng  
RAM nội có địa chỉ đầu là 40H.  
ORG 0000H  
MOV R0,#30H  
MOV R1,#40H  
MOV R2,#10  
lap: MOV A,@R0  
MOV @R1,A  
INC R0  
INC R1  
DJNZ R2,lap  
END  
2.30 Viết CT chuyển một chuỗi dữ liệu gồm 100 byte trong RAM ngoài có địa chỉ đầu là 2000H  
đến vùng RAM ngoài có địa chỉ đầu là 4000H.  
2.31 Viết CT chuyển một chuỗi dữ liệu gồm 10 byte trong RAM nội có địa chỉ đầu là 30H đến vùng  
RAM ngoài có địa chỉ đầu là 4000H.  
ORG 0000H  
MOV R0,#30H  
MOV DPTR,#4000H  
MOV R3,#10  
lap: MOV A,@R0  
MOVX @DPTR,A  
INC R0  
INC DPTR  
DJNZ R3,lap  
END  
2.32 Viết CT chuyển một chuỗi dữ liệu gồm 10 byte trong RAM ngoài có địa chỉ đầu là 5F00H đến  
vùng RAM nội có địa chỉ đầu là 40H.  
2.33 Cho một chuỗi dữ liệu gồm 20 byte liên tiếp trong RAM nội, bắt đầu từ địa chỉ 20H. Hãy viết  
CT lần lượt xuất các dữ kiệu này ra Port 1.  
2.34 Giả sử Port 1 được nối đến một thiết bị phát dữ liệu (ví dụ như 8 nút nhấn). Hãy viết CT nhận  
liên tiếp 10 byte dữ liệu từ thiết bị phát này và ghi vào 10 ô nhớ (RAM nội) liên tiếp bắt đầu từ ô  
nhớ 50H.  
ORG 0000H

MOV R0,#50H  
MOV R4,#10  
loop: MOV @R0,P1  
INC R0  
DJNZ R4,loop  
END  
Tạo trễ (delay)2.35 Viết CT con delay 100µs, biết rằng thạch anh (xtal) dùng trong hệ thống là:  
a. 12 MHz  
Với thạch anh 12MHz thì chu ký máy TM = 1µs. Do đó, 100 µs = 100 TM  
ctcondelay:  
MOV R2,#50  
DJNZ R2,$  
RET  
b. 6 MHz  
Với thạch anh 6MHz thì chu ký máy TM = 2µs. Do đó, 100 µs = 50 TM  
ctcondelay:  
MOV R2,#25  
DJNZ R2,$  
RET  
2.36 Viết CT con delay 100ms, biết rằng thạch anh (xtal) dùng trong hệ thống là:  
a. 12 MHz  
Với thạch anh 12MHz thì chu ký máy TM = 1µs. Do đó, 100 ms = 100000 TM  
ctcondelay:  
MOV R3,#200  
lap: MOV R2,#250  
DJNZ R2,$  
DJNZ R3,#lap  
RET  
b. 11,0592 MHz  
2.37 Viết CT con delay 1s, biết rằng thạch anh (xtal) dùng trong hệ thống là:  
a. 12 MHz  
b. 24 MHz  
Tạo xung2.38 Viết CT tạo một xung dương (\_⎡⎤\_) tại chân P1.0 với độ rộng xung 1ms, biết rằng xtal  
là 12 MHz.  
CÁCH 1:  
ORG 0000H

CLR P1.0  
SETB P1.0  
MOV R3,#2  
lap: MOV R2,#250  
DJNZ R2,$  
DJNZ R3,lap  
CLR P1.0  
END  
CÁCH 2:  
ORG 0000H  
CLR P1.0  
SETB P1.0  
ACALL delay1ms  
CLR P1.0  
SJMP ketthuc  
delay1ms:  
MOV R3,#2  
lap: MOV R2,#250  
DJNZ R2,$  
DJNZ R3,lap  
RET  
ketthuc:  
NOP  
END  
2.39 Viết CT tạo chuỗi xung vuông có f = 100 KHz tại chân P1.1 (Xtal 12 MHz).  
ORG 0000H  
lap: CPL P1.1  
NOP  
NOP  
SJMP lap  
END  
2.40 Viết CT tạo chuỗi xung vuông có f = 100 KHz và có chu kỳ làm việc D = 40% tại chân P1.2  
(Xtal 12 MHz).  
ORG 0000H  
lap: SETB P1.2  
NOP  
NOP  
NOP  
CLR P1.2  
NOP

NOP  
NOP  
SJMP lap  
END  
2.41 Viết CT tạo chuỗi xung vuông có f = 10 KHz tại chân P1.3 (Xtal 24 MHz).  
ORG 0000H  
lap: CPL P1.3  
ACALL delay50  
SJMP lap  
delay50:  
MOV R4,#25  
DJNZ R4,$  
RET  
END  
2.42 Viết CT tạo chuỗi xung vuông có f = 10 KHz và có chu kỳ làm việc D = 30% tại chân P1.3  
(Xtal 24 MHz).  
ORG 0000H  
lap: SETB P1.3  
ACALL delay30  
CLR P1.3  
ACALL delay70  
SJMP lap  
delay30:  
MOV R4,#15  
DJNZ R4,$  
RET  
delay70:  
MOV R4,#35  
DJNZ R4,$  
RET  
END  
2.43 Viết CT tạo chuỗi xung vuông có f = 10 Hz tại chân P1.4 (Xtal 12 MHz).  
2.44 Viết CT tạo chuỗi xung vuông có f = 10 Hz và có chu kỳ làm việc D = 25% tại chân P1.5 (Xtal  
12 MHz).  
Các phép toán2.45 Cho một chuỗi số 8 bit không dấu trong RAM nội gồm 10 số bắt đầu từ ô nhớ 30H. Hãy viết  
CT con cộng chuỗi số này và ghi kết quả vào ô nhớ 2FH (giả sử kết quả nhỏ hơn hoặc bằng 255).  
ORG 0000H  
MOV R0,#30H  
CLR A

MOV R2,#10  
lap: ADD A,@R0  
INC R0  
DJNZ R2,lap  
MOV 2FH,A  
END  
2.46 Cho một chuỗi số 8 bit không dấu trong RAM nội gồm 10 số bắt đầu từ ô nhớ 30H. Hãy viết  
CT con cộng chuỗi số này và ghi kết quả vào ô nhớ 2EH:2FH (ô nhớ 2EH chứa byte cao của kết  
quả và ô nhớ 2FH chứa byte thấp của kết quả).  
ORG 0000H  
MOV R0,#30H ;địa chỉ bắt đầu

MOV R2,#10 ;số lần lặp

CLR A ;byte thấp của kết quả

MOV 2EH,#0 ;byte cao của kết quả

CLR C

lap: ADD A,@R0

JNC boqua

INC 2EH

boqua:INC R0

DJNZ R2,lap

MOV 2FH,A

END

2.47 Cho một chuỗi số 16 bit không dấu trong RAM nội gồm 10 số bắt đầu từ ô nhớ 30H theo  
nguyên tắc ô nhớ có địa chỉ nhỏ hơn chứa byte cao và ô nhớ có địa chỉ lớn hơn chứa byte thấp. (Ví  
dụ: byte cao của số 16 bit đầu tiên được cất tại ô nhớ 30H và byte thấp của số 16 bit đầu tiên được  
cất tại ô nhớ 31H). Hãy viết CT con cộng chuỗi số này và cất kết quả vào ô nhớ 2EH:2FH.  
2.48 Viết CT con lấy bù 2 số 16 bit chứa trong R2:R3.  
So sánh2.49 Cho hai số 8 bit, số thứ 1 chứa trong (30H), số thứ 2 chứa trong (31H). Viết CT con so sánh  
hai số này. Nếu số thứ 1 lớn hơn hoặc bằng số thứ 2 thì set cờ F0, nếu ngược lại thì xóa cờ F0.  
ORG 0000H  
MOV A,30H

CJNE A,31H,ke hoac CJNE A,31H,$+3

CJNE A,31H,ke hoac CJNE A,31H,$+3

ke: JNC lonhoacbang JNC lonhoacbang

CLR F0  
SJMP tiep  
lonhoacbang:  
SETB F0  
tiep: NOP  
END

2.50 Cho hai số 16 bit, số thứ 1 chứa trong (30H):(31H), số thứ 2 chứa trong (32H):(33H). Viết CT  
con so sánh hai số này. Nếu số thứ 1 lớn hơn hoặc bằng số thứ 2 thì set cờ F0, nếu ngược lại thì xóa  
cờ F0.  
2.51 Cho một chuỗi ký tự dưới dạng mã ASCII trong RAM nội, dài 20 byte, bắt đầu từ địa chỉ 50H.  
Viết CT xuất các ký tự in hoa có trong chuỗi này ra Port 1. Biết rằng mã ASCII của ký tự in hoa là  
từ 65 (chữ A) đến 90 (chữ Z).  
ORG 0000H  
MOV R0,#50H  
MOV R3,#20

|  |  |  |  |
| --- | --- | --- | --- |
| lap: | MOV A,@R0 CJNE A,#65,ke | hoac | CJNE A,#65,$+3 |
| ke: | JC boqua  CJNE A,#91,ke2 |  | JC boqua  CJNE A,#91,$+3 |
| ke2: | JNC boqua  MOV P1,A |  | JNC boqua |

boqua:INC R0  
DJNZ R3,lap  
END  
2.52 Viết CT nhập một chuỗi ký tự từ Port 1 dưới dạng mã ASCII và ghi vào RAM ngoài, bắt đầu  
từ địa chỉ 0000H. Biết rằng chuỗi này kết thúc bằng ký tự CR (có mã ASCII là 0DH) và ghi cả ký  
tự này vào RAM.  
ORG 0000H  
MOV DPTR,#0000H  
tiep: MOV A,P1  
MOVX @DPTR,A  
INC DPTR  
CJNE A,#0DH,tiep  
END  
2.53 Viết CT nhập một chuỗi ký tự từ Port 1 dưới dạng mã ASCII và ghi vào RAM ngoài, bắt đầu  
từ địa chỉ 0000H. Biết rằng chuỗi này kết thúc bằng ký tự CR (có mã ASCII là 0DH) và không ghi  
ký tự này vào RAM.  
ORG 0000H  
MOV DPTR,#0000H

|  |  |
| --- | --- |
| lap: | MOV A,P1 CJNE A,#0DH,tiep |

SJMP ketthuc  
tiep: MOVX @DPTR,A  
INC DPTR  
SJMP lap  
ketthuc:  
NOP

2.54 Viết CT nhập một chuỗi ký tự từ Port 1 dưới dạng mã ASCII và ghi vào RAM ngoài, bắt đầu  
từ địa chỉ 0000H. Biết rằng chuỗi này kết thúc bằng ký tự CR (có mã ASCII là 0DH) và không ghi  
ký tự này vào RAM mà thay bằng ký tự null (có mã ASCII là 00H).  
2.55 Cho một chuỗi ký tự dưới dạng mã ASCII trong RAM nội, dài 20 byte, bắt đầu từ địa chỉ 50H.  
Viết CT đổi các ký tự in hoa có trong chuỗi này thành ký tự thường. Biết rằng mã ASCII của ký tự  
thường bằng mã ASCII của ký tự in hoa cộng thêm 32.  
2.56 Cho một chuỗi ký tự số dưới dạng mã ASCII trong RAM nội, dài 20 byte, bắt đầu từ địa chỉ  
50H. Viết CT đổi các ký tự số này thành mã BCD. Biết rằng mã ASCII của các ký tự số là từ 30H  
(số 0) đến 39H (số 9).  
Sử dụng lệnh nhảy có điều kiện2.57 Cho một chuỗi dữ liệu dưới dạng số có dấu trong RAM ngoài, dài 100 byte, bắt đầu từ địa chỉ  
0100H. Viết CT lần lượt xuất các dữ liệu trong chuỗi ra Port 1 nếu là số dương (xem số 0 là dương)  
và xuất ra Port 2 nếu là số âm.  
ORG 0000H  
MOV DPTR,#0100H  
MOV R4,#100

|  |  |
| --- | --- |
| lap: | MOVX A,@DPTR JNB ACC.7,duong |

MOV P2,A  
SJMP tiep  
duong:MOV P1,A  
tiep: INC DPTR  
DJNZ R4,lap  
END  
2.58 Cho một chuỗi dữ liệu dưới dạng số có dấu trong RAM ngoài, bắt đầu từ địa chỉ 0100H và kết  
thúc bằng số 0. Viết CT lần lượt xuất các dữ liệu trong chuỗi ra Port 1 nếu là số dương và xuất ra  
Port 2 nếu là số âm.  
ORG 0000H  
MOV DPTR,#0100H

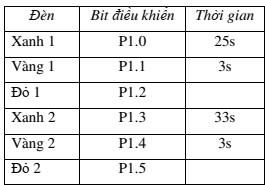
|  |  |
| --- | --- |
| lap: | MOVX A,@DPTR JNB ACC.7,duong |

MOV P2,A  
SJMP tiep  
duong:MOV P1,A  
tiep: INC DPTR  
CJNE A,#0,lap  
END  
2.59 Cho một chuỗi dữ liệu dưới dạng số không dấu trong RAM ngoài, bắt đầu từ địa chỉ 0100H và  
độ dài chuỗi là nội dung ô nhớ 00FFH. Viết CT đếm số số chẵn (chia hết cho 2) có trong chuỗi và  
cất vào ô nhớ 00FEH.  
HD: Số chẵn có LSB=0

2.60 Cho một chuỗi dữ liệu dưới dạng số không dấu trong RAM ngoài, bắt đầu từ địa chỉ 0100H và  
độ dài chuỗi là nội dung ô nhớ 00FFH. Viết CT ghi các số chẵn (xem số 0 là số chẵn) có trong  
chuỗi vào RAM nội bắt đầu từ địa chỉ 30H cho đến khi gặp số lẻ thì dừng.  
2.61 Viết CT con có nhiệm vụ lấy 1 byte từ 1 chuỗi data gồm 20 byte cất trong Ram ngoài bắt đầu  
từ địa chỉ 2000H và xuất ra Port1. Mỗi lần gọi CT con chỉ xuất 1 byte, lần gọi kế thì xuất byte kế  
tiếp, lần gọi thứ 21 thì lại xuất byte đầu, ...

**TIMER**

3.1 Viết CT con mang tên DELAY500 có nhiệm vụ tạo trễ 0,5ms dùng Timer. (Xtal 6MHz).  
…  
MOV TMOD,#00000010B  
…  
delay500:  
MOV TL0,#-250  
SETB TR0  
JNB TF0,$  
CLR TR0  
CLR TF0  
RET  
3.2 Viết CT con mang tên DELAY10 có nhiệm vụ tạo trễ 10ms dùng Timer. (Xtal 12MHz).  
…  
MOV TMOD,#00000001B  
…  
delay10:  
MOV TH0,#HIGH(-10000)  
MOV TL0,#LOW(-10000)  
SETB TR0  
JNB TF0,$  
CLR TR0  
CLR TF0  
RET  
3.3 Dùng CT con DELAY500 (bài 3.1) để viết CT tạo sóng vuông f=1KHz tại P1.0.  
3.4 Dùng CT con DELAY10 (bài 3.2) để viết CT tạo sóng vuông f=50Hz tại P1.1.  
3.5 Dùng CT con DELAY500 (bài 3.1) để viết CT tạo sóng vuông f=500Hz (D=25%) tại P1.2.  
3.6 Dùng CT con DELAY10 (bài 3.2) để viết CT tạo sóng vuông f=20Hz (D=20%) tại P1.3.  
3.7 Viết CT dùng Timer tạo sóng vuông f=500Hz tại P1.4. (Xtal 12MHz).  
3.8 Viết CT dùng Timer tạo sóng vuông f=20KHz tại P1.5. (Xtal 24MHz).  
3.9 Viết CT dùng Timer tạo 2 sóng vuông có cùng f= 1KHz tại P1.6 và P1.7. Biết rằng sóng vuông  
tại P1.7 chậm pha hơn sóng vuông tại P1.6 100(s. (Xtal 12MHz).  
3.10 Viết CT dùng Timer điều khiển đèn giao thông tại một giao lộ. Cho biết rằng:



Đèn sáng khi bit điều khiển bằng 0.

**SERIAL PORT**

4.1 Viết CT đọc 1 chuỗi data chứa trong RAM nội từ địa chỉ 30H đến 50H và xuất ra 1 thiết bị (ví  
dụ như màn hình tinh thể lỏng LCD) được nối với port nối tiếp của 8051 (chế độ UART 8 bit, 2400  
baud). Cho Xtal 11,059 MHz.  
ORG 0000H  
MOV TMOD,#00100000B  
MOV SCON,#01010010B  
MOV TH1,#-12  
SETB TR1  
MOV R0,#30H  
lap: MOV A,@R0  
JNB TI,$  
CLR TI  
MOV SBUF,A  
INC R0  
CJNE R0,#51H,lap  
END  
4.2 Viết CT nhận 1 chuỗi data từ 1 thiết bị ngoài (ví dụ như máy đọc mã vạch) nối với 8051 qua  
port nối tiếp (chế độ UART 8 bit, 4800 baud) và ghi data vào RAM nội từ địa chỉ 40H. Biết rằng  
chuỗi data gồm 20 byte và Xtal 11,059MHz.  
ORG 0000H  
MOV TMOD,#00100000B  
MOV SCON,#01010010B  
MOV TH1,#-6  
SETB TR1  
MOV R0,#40H  
MOV R2,#20

|  |  |
| --- | --- |
| lap: | JNB RI,$ CLR RI |

MOV A,SBUF  
MOV @R0,A

INC R0  
DJNZ R2,lap  
END  
4.3 Viết CT lấy 1 chuỗi data chứa trong RAM ngoài bắt đầu từ địa chỉ 2000H và xuất ra 1 thiết bị  
được nối với port nối tiếp của 8051 (chế độ UART 8 bit, 1200 baud). Chuỗi kết thúc bởi ký tự EOT  
(có mã ASCII là 04H) và ký tự này cũng được xuất ra (Xtal 11,059 MHz).  
4.4 Làm lại bài 4.3 nhưng không xuất ký tự EOT.  
4.5 Viết CT nhận 1 chuỗi data từ 1 thiết bị ngoài nối với 8051 qua port nối tiếp (chế độ UART 8 bit,  
9600 baud) và ghi data vào RAM ngoài bắt đầu từ địa chỉ 4000H. Chuỗi data bắt đầu bằng ký tự  
STX (02H) và kết thúc bằng ký tự ETX (03H). Không ghi 2 ký tự này vào RAM. Cho Xtal  
11,059MHz.  
4.6 Viết CT con mang tên XUAT có nhiệm vụ lấy 1 chuỗi data chứa trong RAM ngoài xuất ra port  
nối tiếp ở chế độ UART 9 bit. Bit thứ 9 là bit parity chẵn. Chuỗi data kết thúc bằng ký tự null  
(00H). CT gọi CT con XUAT sẽ đặt địa chỉ bắt đầu của chuỗi vào DPTR trước khi gọi CT con  
XUAT. Giả sử port nối tiếp đã được khởi động.  
4.7 Viết CT con mang tên NHAP có nhiệm vụ nhập 1 chuỗi data gồm 30 byte từ port nối tiếp ở chế  
độ UART 9 bit, bit thứ 9 là bit parity lẻ. Nếu data nhận được không bị lỗi thì ghi vào 1 vùng nhớ  
của RAM nội, nếu bị lỗi thì không ghi. CT gọi CT con NHAP sẽ đặt địa chỉ đầu của vùng nhớ vào  
thanh ghi R0 trước khi gọi CT con NHAP. Giả sử port nối tiếp đã được khởi động.

5. INTERRUPT

5.1 Viết CT dùng ngắt Timer để tạo sóng vuông f=2KHz tại P1.7. (Xtal 12MHz).  
5.2 Viết CT dùng ngắt Timer để tạo sóng vuông f=200Hz tại P1.6. (Xtal 12MHz).  
5.3 Viết CT dùng ngắt Timer để tạo đồng thời 2 sóng vuông 1KHz và 50Hz tại P1.0 và P1.1. (Xtal  
6MHz)  
5.4 Viết CT lấy 1 chuỗi data chứa trong Ram ngoài bắt đầu từ địa chỉ 6200H đến địa chỉ 62FFH và  
xuất ra Port1, mỗi lần xuất cách nhau 50ms. Sử dụng ngắt Timer. Xtal 12MHz.  
5.5 Viết CT nhập data từ thiết bị ngoài kết nối với 8051 qua Port1, mỗi lần nhập cách nhau 5s, data  
nhập về được ghi vào vùng Ram nội bắt đầu từ địa chỉ 50H đến địa chỉ 5FH. Biết rằng sau khi ghi  
vào ô nhớ cuối cùng thì trở lại ghi vào ô nhớ đầu. Sử dụng ngắt Timer. Xtal 12MHz.  
5.6 Viết CT phát liên tục chuỗi số từ 0 đến 9 ra port nối tiếp theo chế độ UART 8 bit, 2400 baud. Sử  
dụng ngắt serial. Xtal 12MHz.  
5.7 Viết CT chờ nhận data từ 1 thiết bị ngoài gởi đến 8051 qua port nối tiếp (chế độ UART 8 bit,  
19200 baud). Nếu nhận được ký tự STX (02H) thì bật sáng LED, nếu nhận được ký tự ETX (03H)  
thì tắt LED, biết rằng LED được điều khiển bằng ngõ P1.3 (LED sáng khi bit điều khiển bằng 1).  
Sử dụng ngắt serial. Xtal 11,059MHz.  
5.8 Viết CT chờ nhận 1 xung cạnh xuống đưa vào chân /INT0 (P3.2), khi có xung thì nhập data từ  
Port1 và phát ra port nối tiếp ở chế độ UART 9 bit 4800 baud, bit thứ 9 là bit parity lẻ. Xtal 6MHz.  
5.9 Viết CT đếm số xung đưa vào chân /INT1 (P3.3) và điều khiển relay thông qua chân P3.0 (relay  
đóng khi P3.0 bằng 1), cất số đếm vào ô nhớ 40H của Ram nội, nếu số đếm chưa đến 100 thì đóng  
relay, nếu số đếm đạt 100 thì ngắt relay.