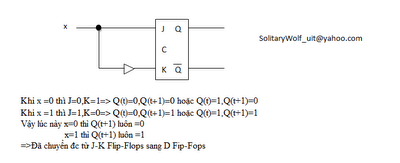
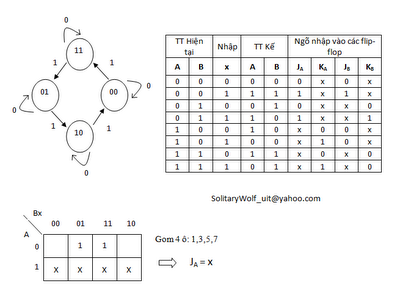
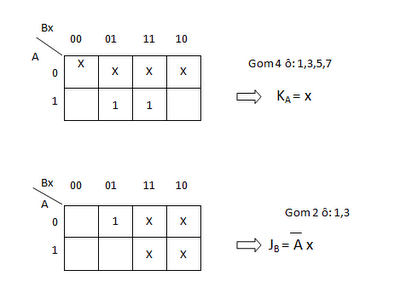
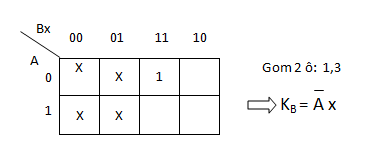
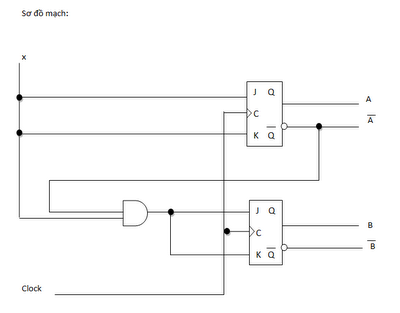
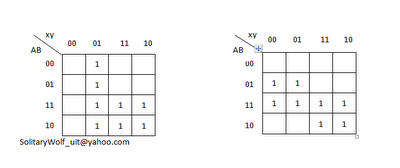
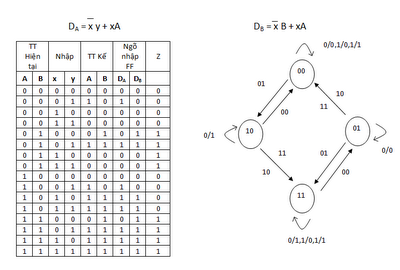
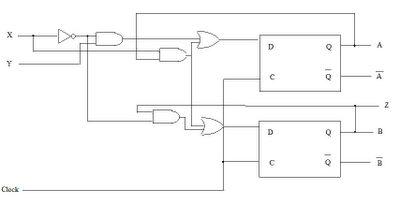
Câu 1: Chứng minh JK Flip-Flop có thể chuyển sang D fiplop.  
  
[](http://3.bp.blogspot.com/-8qW5zbFJ35k/T4fa6WCY-kI/AAAAAAAAAIA/Xt5Ut25BGzQ/s1600/Snapshot_2012-04-09_163544.png)Câu 2:Thiết kế mạch tuần tự dùng mạch lật JK.Khi ngõ nhập x=0,trạng thái mạch lật ko đổi.Khi x=1,dãy trạng thái là 11,01,10,00 và lặp lại.  
  
  
[](http://2.bp.blogspot.com/-CvqSwYOIs_w/T4KuuoqPhMI/AAAAAAAAAGY/6Bi_AVS2H-g/s1600/Snapshot_2012-04-09_164018.png)  
[](http://2.bp.blogspot.com/-8EcznIaJ0Yg/T4KvKvUjfoI/AAAAAAAAAGk/WruwYqvKKik/s1600/Snapshot_2012-04-09_164222.png)  
[](http://3.bp.blogspot.com/-456a9PGQ3ZY/T4Kvf0AbtsI/AAAAAAAAAGw/XYmFckHvGD4/s1600/Snapshot_2012-04-09_164319.png)  
[](http://2.bp.blogspot.com/-2Ex-0MERUfw/T4KwDwRyLJI/AAAAAAAAAG8/wOHyi49-9tg/s1600/Snapshot_2012-04-09_164627.png)

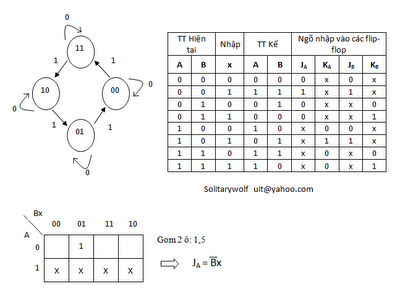
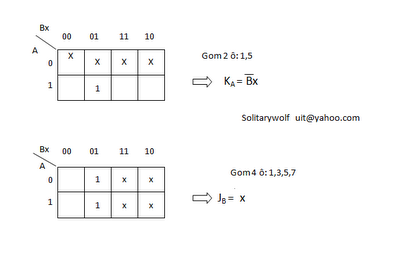
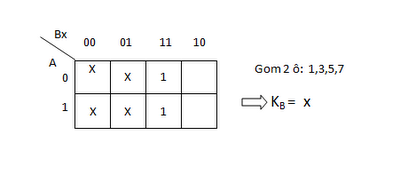
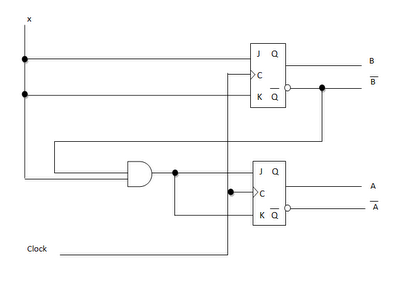
Câu 3: 1 mạch tuần tự gồm 2 D Flip-Flop A,B,2 ngõ nhập x,y một ngõ xuất z.Phương trình các ngõ nhập vào các Flip-Flop và ngõ xuất mạch như sau:  
DA=not x.y+x.A  
DB=not x.B+x.A  
Z=B  
a/Vẽ lược đồ luận lí mạch.  
b/Lập bảng trạng thái.  
[](http://2.bp.blogspot.com/-KUAnSTwVWVU/T4KxNObVRCI/AAAAAAAAAHI/A4Bx2jDLSjs/s1600/Snapshot_2012-04-09_165124.png)  
  
[](http://1.bp.blogspot.com/-MPNxoRyHbYI/T4VUoG296pI/AAAAAAAAAHs/Pb0QeLBDbyo/s1600/Snapshot_2012-04-09_165155.png)  
  
[](http://2.bp.blogspot.com/-Ox0QOkUFG9o/T4f66I20QXI/AAAAAAAAAIY/zYczjkehFMc/s1600/Snapshot_2012-04-13_170647.png)

Mình giải thích tí:cái lược đồ luận lí(hay trong sách nó ghi là sơ đồ mạch cũng như nhau thôi:D)  
Còn về lược đồ trạng thái:  
-ở trạng thái 0/0 nếu x/y mà là 0/0,1/0 hay 1/1 thì nó vẫn giữ nguyên trạng thái ban đầu là 0/0.Tương tự cho trạng thái 0/1,1/1 và 1/0.  
-ở trạng thái 0/0 nếu x/y=0/1 thì trạng thái chuyển từ 0/0 sang 1/0.Nói chung là khi có bảng trạng thái thì việc các bạn xác định đc lược đồ trạng thái là vô cùng dễ:D.

Bonus

câu 1:Thiết kế mạch đếm giảm 2 bit. Đây là mạch tuần tự có 2 J-K flip-flop và 1

ngõ nhập x. Khi x=0, trạng thái mạch lật không đổi. Khi x=1, dãy trạng thái là 11,

10, 01, 00 và lặp lại. [](http://4.bp.blogspot.com/-FSVQ2bB0jBY/T4gVn2j2CvI/AAAAAAAAAIw/ZAgL05xD-Lo/s1600/Snapshot_2012-04-13_190059.png)  
  
[](http://2.bp.blogspot.com/-OL1sb4nl818/T4gWF1PQmdI/AAAAAAAAAI8/qyYEl3lfT-k/s1600/Snapshot_2012-04-13_190307.png)  
[](http://2.bp.blogspot.com/-ip4m2pgRWFI/T4gWufA-dXI/AAAAAAAAAJI/sZMDK5p9W9c/s1600/Snapshot_2012-04-13_190446.png)Sơ đồ mạch  
[](http://1.bp.blogspot.com/-5osGE9Omexw/T4gW5EAG7qI/AAAAAAAAAJU/LOREVHrRnIU/s1600/Snapshot_2012-04-13_190508.png)

Chương VI

1/có 1 kí tự ->8 bit =>bộ nhớ chính chứa 2­24\*8 bit

a.1 ô 8 bit => có (2­24\*8)/8 = 224 ô vậy trường địa chỉ sẽ cần 24 bit

b.1 ô 16 bit =>có (2­24\*8)/16 = 8388608 ô vậy trường địa chỉ sẽ cần 23 bit

c.1 ô 32 bit => có (2­24\*8)/32 = 4194304 ô vậy trường địa chỉ sẽ cần 22 bit

2/

a.7 lệnh có hai địa chỉ 15 bit và một số hiệu thanh ghi 3 bit  
opcode chiếm 36 - (2\*15 + 3) = 3 bit  
3bit mã hóa đc 8 lệnh  
yêu cầu 7 lệnh =>cần 3 bit để mã hóa  
và 7 lệnh từ 000 đến 110 còn 111 đc xử lí đặc biệt.

36 bit

000 xxxxxxxxxxxxxxx yyyyyyyyyyyyyyy zzz

001 xxxxxxxxxxxxxxx yyyyyyyyyyyyyyy zzz

……………………………………………………………….

110 xxxxxxxxxxxxxxx yyyyyyyyyyyyyyy zzz

b. 500 lệnh có một địa chỉ 15 bit và một số hiệu thanh ghi 3 bit  
opcode chiếm 36 - (15 + 3 ) = 18 bit.  
18 bit mã hóa đc 218 lệnh  
yêu cầu 500 lệnh cần 9 bit để mã hóa   
và 500 lệnh từ 111111111 000000000 đến 111111111 111110011 , phần còn lại đc xử lí đặc biệt.

36 bit

111111111 000000000 xxxxxxxxxxxxxxx yyy

111111111 000000001 xxxxxxxxxxxxxxx yyy

……………………………………………………………………

111111111 111110011 xxxxxxxxxxxxxxx yyy

c. 50 lệnh không có địa chỉ hoặc thanh ghi  
opcode chiếm 36 - 0 -0 = 36 bit  
36 bit mã hóa đc 236 lệnh  
yêu cầu 50 lệnh cần 6 bit để mã hóa   
---> 50 lệnh 1111…1111 000000 đến 111…1111 110001 phần còn lại đc xử lí đặc biệt.

36 bit

111111111111111111111111111111 000000

111111111111111111111111111111 000000

……………………………………………………………………….

111111111111111111111111111111 110001

3/

a. 4 lệnh có ba thanh ghi  
4 lệnh cần 2 bit, 3 thanh ghi mỗi thanh 3 bit => 9 bit  
=> tổng cộng cần 2 + 3 \* 3 = 11 bit  
vậy thiết kế đc.  
b. 255 lệnh có hai thanh ghi  
255 lệnh cần 8 bit, 2 thanh ghi mỗi thanh 3 bit=> 6 bit  
--> tổng cộng cần 8 + 6 = 14 bit   
vậy ko thiết kế đc.  
c. 2048 lệnh không có thanh ghi  
2048 lệnh cần 11 bit, ko có thanh ghi nên ko cần thêm bit.11 bit   
vậy thiết kế dc.

Bonus

4/Cho biết kết quả các thanh ghi AX, BX sau tập lệnh

MOV AX,5070 H //AX <- <5070H>

MOV BX,4670 H //BX <- <4670H>

ADD AX,BX //AX <- AX + BX

INC AX//AX=AX+1

BX=<4670H>,AX =<5070H>+<4670H>+1

MOV AX,7070 H //AX<- <7070H>

MOV BX,E6F0 H //BX<-<E6F0H>

ADD AX,BX //AX<-AX+BX

INC AX//AX<- AX+1

BX =<E6F0H>,AX = <7070H>+<E6F0H>+1

5/ Cho đoạn lệnh sau, sau khi thực hiện lệnh cho biết kết quả của các thanh ghi AX, BX, CX, DX.

MOV AX, 34 H//AX<- <34H>

MOV BX, 10 H //BX<- <10H>

DIV BX

AX=<34H>/<10H>,DX=phần dư <34H>/<10H>

MOV AX, 14 H //AX<-<14H>

MOV BX, 12 H //BX<-<12H>

MUL BX

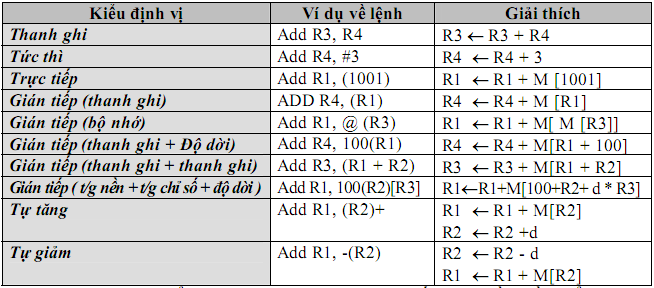
AX=16 bit thấp của <14H>\*<12H>,DX=16 bit cao của <14H>\*<12H>,BX=<12H>

6/Việc thi hành lệnh mã máy được thực thi ra sao? giải thích việc thực thi đoạn chương trình sau:

Đầu tiên là đọc lệnh=>giải mã=>tìm nạp bộ nhớ cần dùng cho lệnh=>thực thi lệnh=>ghi lại kết quả vào vùng nhớ.

MOV R4, #24 //R4<-24

ADD R4, (R1) //R4 <- R4 + mem(R1)



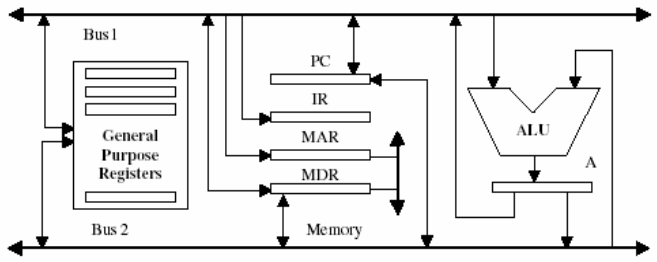
**Những chữ số trong biểu diễn bằng 7 đoạn sẽ như sau:**  
0 1 2 3 4 5 6 7 8 9   
  
input cho mạch giải mã 7 đoạn là 4 input tương ứng với 4 bit biểu diễn cho giá trị thập phân từ 0 đến 9 ( và 10 đến 15 là những trường hợp "don't care"), và mỗi output là một của mỗi đoạn cần hiển thị  
a. Lập bảng chân trị vài trình bày bảng đồ K-map cho đoạn giữa (đoạn phân biệt giữa số 0 và số 8), thu gọn thành tổng của tích và vẽ mạch cho đoạn này (3 đ)  
b. Lập bảng chân trị và trình bày bảng đồ K-map cho đoạn đáy (đoạn tồn tại trong tất cả các chữ số ngoại trừ 1, 4 và 7), thu gọn thành tổng của tích và vẽ mạch cho đoạn này(3 đ) 

Có tất cả 10 số tư 0 - 9. ==> cần có 2^4 bit vào (vì 2^3 = 8 ko đủ).   
Có tất cả 7 đầu ra thể hiện cho 7 đoạn (7 led). Mỗi chân ra thể hiện 1 đoạn (led).  
a/ Làm như một bài giải mã bình thường thôi. Ví dụ: 001 là 1 thì ở led 7 đoạn bạn cho những đoạn nào sáng để nó thành số 1 (số 1 thì cần 2 led sáng , tức = 1). Ở câu A chỉ cần làm những số có đoạn gạch ngang ở giữa như đề đã nói: 2,3,4,5,6,8,9. Mấy số kia khỏi làm.  
b/ Làm như câu trên nhưng chỉ cần làm ở những số nào có đoạn gạch ngang ở chân thôi. (Tất cả các số trừ những số mà đề đã nêu ra).

Bạn liên hệ thức tế nhé về led 7 đoạn nhé. Hay xem hình của nó   
Nhìn hình bạn thấy thật ra led 7 đoạn cơ bản là 7 cái đèn nhỏ ở trong đó (nói vậy cho dể hiểu). Nếu bạn muốn nó hiện ra số mấy thì cho những cái đèn nhỏ tương ứng trong đó sáng lên. Thì chúng ta quy định sáng là 1, còn tắt là 0; Ví dụ bạn muốn cho nó hiện số 1 thì bạn cho 2 cái đèn nhỏ ở bên phải sáng lên tạo thành một đường thẳng đứng ==> số 1. Ví dụ muốn nó hiện số 8 thì cho nó sáng luôn 7 đèn, ... Còn việc lập bảng K-map như thế nào thì bạn cần xem sách.   
Ở câu A thì đề kêu bạn lập K-map cho những số nào có đèn ở giữa sáng. đó là những số nào bạn nhình con led 7 đoạn bạn cũng đoán được mà!

**Bài** .giả sử hiện tạ máy tính có thanh ghi R0=1000, R1=50,R2=150; giá trị tại ô nhớ 200 là 1200. Hãy cho biết giá trị thanh ghi và ô nhớ trên sau khi thực hiện đoạn lệnh sau:  
Store 200(R0),R1 //M[200+R0]   
ADD R2,R1  
LOAD R0,@(R2)  
MOVE R2,R0  
(giả sử lệnh 2 toán hạng theo dạng: LỆNH đích,nguồn. )

- STORE 200(R0), R1 : Ô nhớ: 200(R0) = giá trị R0 +200 = 1000+200= 1200 (ô nhớ số 1200) => Kết quả là lưu giá trị R1=50 vào ô nhớ thứ 1200.  
- ADD R2,R1: R1 = 50, R2=150 => R2 = R2+R1 = 150+50 = 200  
- LOAD R0,@(R2): ta có: @(R2)=50 => R0 = 50;  
- MOVE R2,R0 : R2 = RO = 50

**Bài.** Cho tổ chức máy tính two-bus như hình vẽ. Các thanh ghi đa dụng R0,R1,…,R7.   
  
  
  
Hãy giải thích quá trình thi hành lệnh (các thời hiệu và các vi lệnh thực hiện trong các thời hiệu):   
ADD R1, (R2).   
Biết rằng R1=10, R2=100H và giá trị chứa trong ô nhớ 100H là 500.

***two-bus thì có 4 step***  
Step  
t0 : MAR <-- (R2) // MAR<-100H  
t1: MDR <-- M(MAR) // MDR <-M(100H) = 500  
t2: A <-- (MDR) + R1 //A<-- 500 + 10 = 510  
t3: R1<--A

dạng two-bus có thề viết 1 dạng khác  
Step  
t0 : MAR <-- (R2) // MAR<-100H  
t1: MDR <-- M(MAR) // MDR <-M(100H) = 500  
t2: A<- (MDR) //A <-- 500  
t3: R1 <-- (R1) + A //R1<-- 10+500= 510

***one-bus thì có 5 step***  
Step  
t0 : MAR <-- (R2) // MAR<-100H  
t1: MDR <-- M(MAR) // MDR <-M(100H) = 500  
t2: A <-- (MDR)//A<-- 500  
t3: B <-- (R1) //B<-10  
t4: R1<-- A + B  
  
 ***3-bus***  
Step  
t0 : MAR <-- (R2) // MAR<-100H  
t1: MDR <-- M(MAR) // MDR <-M(100H) = 500  
t2: R1<-- (MDR) + (R1) //A<-- 500 + 10 = 510

*Đây là bài tập phần "Vi tác vụ" của đề thi năm trước! Mình có xem slide về phần này nhưng khó hiểu quá. Mong các bạn giúp đỡ.*

*Mã:*

*Các vi lệnh sau được thực hiện trong hệ thống như hình vẽ. Với mỗi vi lệnh, xác định:*

*• Gía trị các ngõ nhập chọn bus S2, S1, S0.*

*• Thanh ghi và ngõ nhập điều khiển LD (nếu có).*

*• Tác vụ đọc hoặc ghi bộ nhớ (nếu có).*

*• Tác vụ trong mạch cộng luận lý (nếu có)*

*a) AR ← PC*

*b) IR ← M[AR]*

*c) M[AR] ← TR*

*d) AC ← DR, DR ←AC*

a) Để thực hiện việc mov giá trị của thanh ghi PC vào thanh ghi AR thì phải mở đường số 2 của bus, tức là giá trị của S2S1S0 = 010, sau khi mở đường bus số 2 thì thì dữ liệu từ thanh ghi PC tràn vào bus, để ghi giá trị này vào thanh ghi AR thì tín hiệu LD (Load) của thanh ghi AR phải được set lên 1, các thanh ghi còn lại mình ko set giá trị LD = 1 nên dữ liệu ko tràn vào được. Không thực hiện trên bộ nhớ nên ko có tác vụ đọc/ghi. Không thực hiện phép toán nên ko có tác vụ trong mạch cộng luận lý  
  
b) M[AR] bộ nhớ truy cập giá trị tại địa chỉ được lưu trong thanh ghi AR  
Để mov giá trị của ô nhớ tại địa chỉ lưu trong thanh ghi AR vào thanh ghi IR thì mở đường số 7 của bus, tức là S2S1S0 = 111, sau khi mở đường bus số 7 thì dữ liệu từ bộ nhớ tràn vào bus, để ghi giá trị này vào thanh ghi IR thì tín hiệu LD của thanh ghi IR phải được set lên 1. Vì thực hiện thao tác đọc dữ liệu trên bộ nhớ tai địa chỉ lưu trong thanh ghi nên tín hiệu Read của Memory được set lên 1. Không thực hiện phép toán nên ko có tác vụ trong mạch cộng luận lý  
  
c) câu này thì cũng tương tự câu b  
Ghi dữ liệu từ thanh ghi TR vào bộ nhớ có địa chỉ lưu trong thanh ghi AR thì mở đường bus số 6 (S2S1S0 = 110) để dữ liệu trong thanh ghi TR tràn vào bus, sau đó set tín hiệu Write của Memory lên 1 để thực hiện thao tác ghi giá trị  
  
d) câu này thực hiện Swap giá trị của 2 thanh ghi  
Đầu tiên là đưa giá trị của DR vào thanh ghi AC thông qua bộ cộng logic, lúc này để đảm bảo dữ liệu của DR khi đưa vào AC ko đổi thì ta phải lưu giá trị của AC vào thanh ghi nào đó để bảo toàn giá trị của AC (Ví dụ như IR <- AC, mở đường bus số 4 và set LD của IR lên 1). Sau đó clear AC đi, kết hợp với Input (INPR) bằng 0 thì dữ liệu của DR sẽ lưu chính xác ở trong AC. cái này ta không cần mở đường bus (S2S1S0 = 000) thì đã có đường trực tiếp từ DR đi vào bộ cộng  
Thứ 2 là lưu giá trị của AC vào DR: Vì giá trị của AC lưu trong IR nên ta mở đường bus số 5 để dữ liệu của IR tràn vào bus, sau đó set tín hiệu LD của DR kêb 1 để dữ liệu của IR tràn vào DR  
  
Đây là lập luận, em xem xét để trả lời theo yêu cầu đề bài nhé