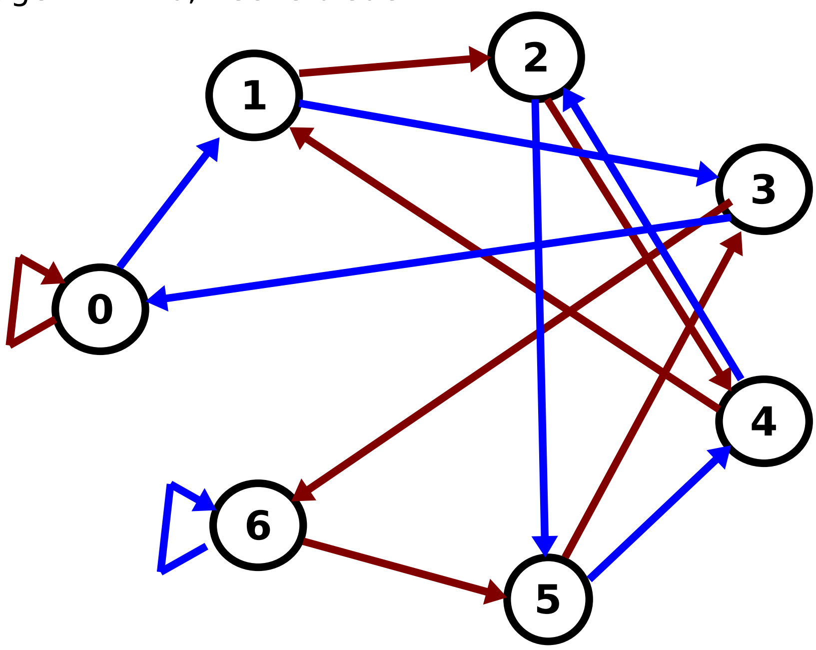
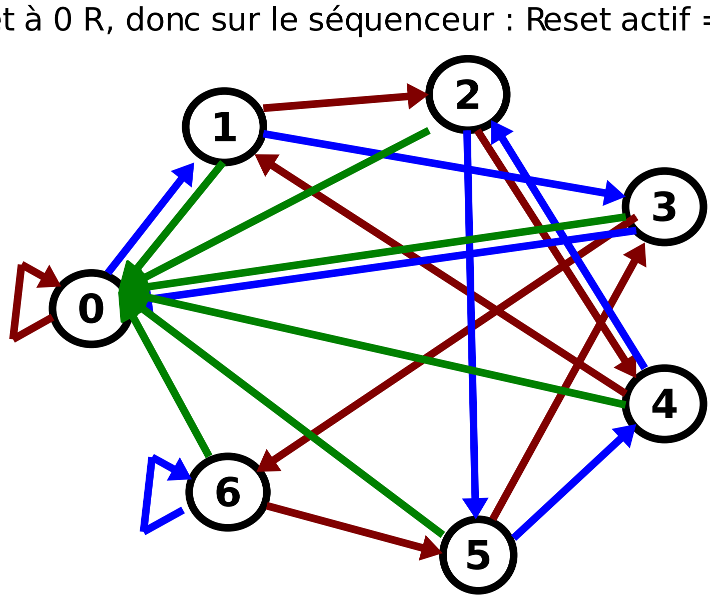
Partiel Électronique Numérique
2002 -2003

1)

2)
Le plus simple pour ce genre de question est de faire un séquenceur qui fournis les valeurs demandés sur les 3 bits R de sortie. En effet, comme est posé la question, il faut un séquenceur qui est 7 états farfelus et après faire un combinatoire qui passe de chacun de ces états en 3 bits lisibles.
Pour ma part, il n'y a qu'un seul bloc séquenceur car les états de sorties du séquenceur correspondent aux états attendu sur R.
3) flèche rouge => E=0, flèche bleue => E=1



4) Le reset remet à 0 R, donc sur le séquenceur : Reset actif => flèche verte



5) On a un séquenceur à 7 états, il nous faut donc 3 bits car : 2^3=8 états possibles

Le codage le plus simple est la conversion binaire de R : 0=000, 1=001, 2=010, 3=011, 4=100 etc..

6) Pour réaliser ce séquenceur à 3 bits, il nous faut 3 bascules JK. On fait d'abord la table des bascules JK (que l'on connaît par coeur) sur front (montant ou pas, on s'en fiche :

CLK	J	K	Qn+1
|	0	0	Qn
|	1	0	1
|	0	1	0
|	1	1	/Qn

Maintenant, on cherche à trouver quel état doit on mettre sur J et K d'une bascule pour qu'au prochain coup,d'horloge, il réagisse correctement suivant Reset et E.
On numérote les bascules JK2, JK1 et JK0 correspondant respectivement aux bits de R.
On fabrique un tableau avec : les entrées E et Reset, les sorties Qn2, Qn1 et Qn0 et enfin, les entrées des bascules J2, K2, J1, K1, J0, K0.
La table de Qni par rapport à E/Reset, on la connaît, il reste plus qu'à compléter les Ji et Ki. (on vas dire Reset actif à 0)

Reset	**E**	**Qn2**	**Qn1**	**Qn0**	**J2**	**K2**	**J1**	**K1**	**J0**	**K0**	**Qn2+1**	**Qn1+1**	**Qn0+1**
0	x	x	x	x	0	1	0	1	0	1	0	0	0
1	0	0	0	0	0	x	0	x	0	x	0	0	0
1	1	0	0	0	0	x	0	x	1	x	0	0	1
1	0	0	0	1	0	x	1	x	x	1	0	1	0
1	1	0	0	1	0	x	1	x	x	0	0	1	1
1	0	0	1	0	1	x	x	1	x	1	1	0	0
1	1	0	1	0	1	x	x	1	1	x	1	0	1
1	0	0	1	1	1	x	x	0	x	1	1	1	0
1	1	0	1	1	0	x	x	1	x	1	0	0	0
1	0	1	0	0	x	1	x	0	1	x	0	0	1
1	1	1	0	0	x	1	1	x	0	x	0	1	0
1	0	1	0	1	x	1	1	x	x	0	0	1	1
1	1	1	0	1	x	0	0	x	x	1	1	0	0
1	0	1	1	0	x	0	x	1	1	x	1	0	1
1	1	1	1	0	x	0	x	0	x	1	1	1	0

On remarque l'on peux dire que si :

- Qn=0 et Qn+1=0 => J=0 et K=x
- Qn=0 et Qn+1=1 => J=1 et K=x
- Qn=1 et Qn+1=0 => J=x et K=1
- Qn=1 et Qn+1=1 => J=x et K=0

maintenant, on réalise les tableaux de karnau pour chaque Ji et Ki en fonction de E, Reset et les Qni.

J2

Qni=>	111	110	100	101	001	000	010	011
11	x	x	x	x	0	0	**1**	0
10	x	x	x	x	0	0	**1**	**1**
00	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0
Res E								

J2= Reset . Qn1 . /(Qn0 + E)

K2

Qni=>	111	110	100	101	001	000	010	011
11	x	0	**1**	0	x	x	x	x
10	x	0	**1**	**1**	x	x	x	x
00	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
01	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Res E								

K2= /Reset + (/Qn1 . /(Qn0 + E))

J1

Qni=>	111	110	100	101	001	000	010	011
11	x	x	**1**	0	**1**	0	x	x
10	x	x	x	**1**	**1**	0	x	x
00	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0
Res E								

J1= Reset . ((/Qn1 . /(Qn0 + E)) + (Qn2 . /Qn0))

K1

Qni=>	111	110	100	101	001	000	010	011
11	x	0	x	x	x	x	**1**	**1**
10	x	**1**	0	x	x	x	**1**	0
00	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
01	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Res E								

K1= /Reset + (/E . Qn2 . Qn1) + (/Qn2 . Qn1 . (/Qn0 + E))

J0

Qni=>	111	110	100	101	001	000	010	011
11	x	x	0	x	x	**1**	**1**	x
10	x	**1**	**1**	x	x	0	x	x
00	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0
Res E								

J0= Reset . ((/E . Qn2) + (E . /Qn2))

K0

Qni=>	111	110	100	101	001	000	010	011
11	x	**1**	x	**1**	0	x	x	**1**
10	x	x	x	0	**1**	x	**1**	**1**
00	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
01	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Res E								

K0= /Reset + (Qn2 . Qn1) + (/Qn1 . Qn0 . ((E . Qn2) + (/E . /Qn2))) + Qn1

Les x sont des états indéterminés, cela signifie cela peut être un 1 ou un 0, cela à le même effet.
Pour les simplifications, ils ont du sens, car on choisi leur état (1 ou 0) suivant comment cela nous arrange (simplifie les équations).

7) le dessin, je vous le laisse

8) S2=Qn2 S1=Qn1 S0=Qn0

9) pas besoin vu notre codage du séquenceur

10) peut être, bascule D plus simple, à voir