

Architecture des systèmes numériques et informatiques

TD 5 : Circuits Séquentiels

Halim Djerroud

5 novembre 2025

Exercice 1 : Bascule RS asynchrone

Une bascule RS est construite avec deux portes NOR croisées.

1. Rappeler le principe de fonctionnement d'une bascule RS.
2. Établir la table de vérité complète avec S , R , Q (état présent) et Q^+ (état suivant).
3. Donner l'équation caractéristique : $Q^+ = f(S, R, Q)$
4. Analyser pas à pas la séquence suivante :

Instant	S	R	Q
t_0	0	0	0
t_1	1	0	?
t_2	0	0	?
t_3	0	1	?
t_4	0	0	?

5. Pourquoi la combinaison $S = R = 1$ est-elle interdite ?
6. Que se passe-t-il si on applique momentanément $S = R = 1$ puis on revient à $S = R = 0$?

Solution : Table de vérité simplifiée :

S	R	Q^+	Observation
0	0	Q	Mémoire
0	1	0	Reset
1	0	1	Set
1	1	X	Interdit

Équation caractéristique : $Q^+ = S + R'Q$ (avec contrainte $SR = 0$)

Séquence :

- $t_0 : Q = 0$ (état initial)
- $t_1 : S = 1, R = 0 \rightarrow Q^+ = 1$ (Set)
- $t_2 : S = 0, R = 0 \rightarrow Q^+ = 1$ (Mémoire)
- $t_3 : S = 0, R = 1 \rightarrow Q^+ = 0$ (Reset)
- $t_4 : S = 0, R = 0 \rightarrow Q^+ = 0$ (Mémoire)

État interdit : $S = R = 1$ force $Q = \overline{Q} = 0$, violation de la complémentarité.

Retour instable : Quand on revient à $S = R = 0$, l'état final dépend des délais de propagation (imprévisible).

Exercice 2 : Bascule RS synchrone

On ajoute une entrée d'horloge H à la bascule RS.

1. Dessiner le schéma logique d'une bascule RS synchrone (utiliser des portes AND).
2. Établir la table de vérité avec l'horloge.
3. Compléter le chronogramme suivant :

Instant	H	S	R	Q
t_0	0	1	0	0
t_1	1	1	0	?
t_2	0	0	1	?
t_3	1	0	1	?
t_4	0	0	0	?

4. Quelle est la différence entre déclenchement sur niveau et sur front ?
5. Proposer une modification pour rendre la bascule sensible au front montant uniquement.

Solution : Table de vérité :

H	S	R	Q^+	Observation
0	X	X	Q	Pas de changement
1	0	0	Q	Mémoire
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	X	Interdit

Chronogramme :

- $t_0 : H = 0 \rightarrow$ pas de changement, $Q = 0$
- $t_1 : H = 1, S = 1, R = 0 \rightarrow Q = 1$ (Set)
- $t_2 : H = 0 \rightarrow$ pas de changement, $Q = 1$
- $t_3 : H = 1, S = 0, R = 1 \rightarrow Q = 0$ (Reset)
- $t_4 : H = 0 \rightarrow$ pas de changement, $Q = 0$

Déclenchement :

- **Niveau** : Active tant que $H = 1$ (peut changer plusieurs fois)
- **Front** : Active uniquement sur transition $0 \rightarrow 1$ ou $1 \rightarrow 0$

Bascule sur front : Utiliser un détecteur de front (circuit avec porte AND et inverseur avec délai).

Exercice 3 : Bascule JK

La bascule JK élimine l'état interdit de la RS en le remplaçant par un mode Toggle.

1. Établir la table de vérité complète de la bascule JK.
2. Vérifier l'équation caractéristique : $Q^+ = J\bar{Q} + \bar{K}Q$
3. Analyser la séquence (bascule déclenchée sur front montant \uparrow) :

Coup d'horloge	J	K	Q (avant)	Q (après)
1	0	0	0	?
2	1	0	?	?
3	0	0	?	?
4	1	1	?	?
5	1	1	?	?
6	0	1	?	?

4. Application : Utiliser une bascule JK en mode Toggle ($J = K = 1$) pour diviser une fréquence par 2.
5. Comment obtenir un diviseur par 4 ? Par 8 ?

Solution : Table de vérité :

J	K	Q^+	Observation
0	0	Q	Mémoire
0	1	0	Reset
1	0	1	Set
1	1	\bar{Q}	Toggle

Vérification de l'équation pour tous les cas :

- $J = 0, K = 0 : Q^+ = 0 \cdot \bar{Q} + 1 \cdot Q = Q$
- $J = 0, K = 1 : Q^+ = 0 \cdot \bar{Q} + 0 \cdot Q = 0$
- $J = 1, K = 0 : Q^+ = 1 \cdot \bar{Q} + 1 \cdot Q = \bar{Q} + Q = 1$

— $J = 1, K = 1 : Q^+ = 1 \cdot \bar{Q} + 0 \cdot Q = \bar{Q}$

Séquence :

Coup	J	K	Q (avant)	Q (après)
1	0	0	0	0 (Mémoire)
2	1	0	0	1 (Set)
3	0	0	1	1 (Mémoire)
4	1	1	1	0 (Toggle)
5	1	1	0	1 (Toggle)
6	0	1	1	0 (Reset)

Diviseur de fréquence :

- $\div 2$: 1 bascule JK en mode Toggle
- $\div 4$: 2 bascules JK en cascade ($\div 2$ puis $\div 2$)
- $\div 8$: 3 bascules JK en cascade
- $\div 2^n$: n bascules JK en cascade

Exercice 4 : Bascule D

La bascule D (Data/Delay) est la plus simple : elle recopie l'entrée D en sortie Q .

1. Établir la table de vérité de la bascule D.
2. Donner l'équation caractéristique : $Q^+ = ?$
3. Comment construire une bascule D à partir d'une bascule JK ?
4. Compléter le chronogramme (front montant) :

H: __|__|__|__|__|__|__|__
 D: __|---|__|----|__
 Q: _____

5. Application : Concevoir un registre 4 bits avec des bascules D.
6. Quelle est la différence entre un latch et un flip-flop ?

Solution : Table de vérité :

D	Q^+
0	0
1	1

Équation caractéristique : $Q^+ = D$ (la plus simple!)

Construction D à partir de JK :

- $J = D$
- $K = \bar{D}$
- Ainsi : si $D = 1 \rightarrow J = 1, K = 0 \rightarrow \text{Set}$
- Si $D = 0 \rightarrow J = 0, K = 1 \rightarrow \text{Reset}$

Chronogramme complété :

H: __|__|__|__|__|__
 t0 t1 t2 t3 t4
 D: __|---|__|----|__
 Q: _____|----|__|----
 ↑ ↑ ↑

Q change sur chaque front montant pour prendre la valeur de D.

Registre 4 bits :

D D D D (entrées parallèles)
 ↓ ↓ ↓ ↓

D D D D

$\leftarrow H$ (horloge commune)
 $Q \ Q \ Q \ Q$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 $Q \ Q \ Q \ Q$ (sorties parallèles)

Latch vs Flip-Flop :

- **Latch** : Transparent quand enable=1 (niveau)
- **Flip-Flop** : Change uniquement sur front d'horloge (front)

Exercice 5 : Conversion entre bascules

On dispose de bascules JK et on souhaite implémenter d'autres types.

1. Bascule D avec JK : donner les connexions.
2. Bascule T (Toggle) avec JK : donner les connexions.
3. Bascule RS avec JK : exprimer J et K en fonction de S et R .
4. Inversement, convertir une bascule D en bascule T :
 - Table de vérité de T : si $T = 0 \rightarrow$ mémoire, si $T = 1 \rightarrow$ toggle
 - Exprimer D en fonction de T et Q
5. Pourquoi la bascule JK est-elle considérée comme "universelle" ?

Solution : 1. D avec JK :

- $J = D$
- $K = \overline{D}$

2. T avec JK :

- $J = T$
- $K = \overline{T}$

3. RS avec JK :

Table d'excitation :

S	R	J	K
0	0	0	0
0	1	0	1
1	0	1	0
1	1	X	X

Solution : $J = S$ et $K = R$ (mais interdit si $SR = 1$)

4. T avec D :

Table de vérité de T :

T	Q	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

On observe : $Q^+ = T \oplus Q$

Donc : $D = T \oplus Q$ (utiliser une porte XOR)

5. Universalité de JK :

La bascule JK peut simuler tous les autres types :

- RS (avec $J = S, K = R$)
- D (avec $J = D, K = \overline{D}$)
- T (avec $J = K = T$)
- Toggle permanent (avec $J = K = 1$)

Exercice 6 : Registre à décalage série

Un registre à décalage de 4 bits est construit avec 4 bascules D en cascade.

1. Dessiner le schéma du registre (entrée série, sortie série).
2. On veut charger la séquence 1011 (MSB en premier). Compléter le tableau :

Cycle	Entrée	Q_3	Q_2	Q_1	Q_0
0	-	0	0	0	0
1	1	?	?	?	?
2	0	?	?	?	?
3	1	?	?	?	?
4	1	?	?	?	?

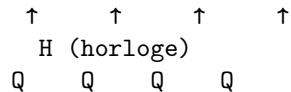
3. Combien de cycles faut-il pour charger complètement le registre ?
4. Comment modifier le circuit pour permettre un chargement parallèle ?
5. Application : Utiliser ce registre pour une transmission série (UART).

Solution : Schéma :

Entrée série

↓

D→Q→D→Q→D→Q→D→Q→ Sortie



Chargement de 1011 :

Cycle	Entrée	Q_3	Q_2	Q_1	Q_0
0	-	0	0	0	0
1	1	1	0	0	0
2	0	0	1	0	0
3	1	1	0	1	0
4	1	1	1	0	1

Nombre de cycles : n cycles pour charger n bits.

Chargement parallèle : Ajouter un multiplexeur 2 : 1 devant chaque bascule D :

- Entrée 0 : donnée parallèle (D_i)
- Entrée 1 : sortie bascule précédente (mode série)
- Sélection : signal LOAD/SHIFT

UART : Le registre à décalage convertit les données parallèles du CPU en transmission série bit par bit sur la ligne TX.

Exercice 7 : Registre à décalage universel

Un registre universel peut fonctionner dans plusieurs modes.

Modes de fonctionnement :

- Mode 0 : Maintien (pas de changement)
- Mode 1 : Décalage à droite
- Mode 2 : Décalage à gauche
- Mode 3 : Chargement parallèle

1. Proposer un schéma bloc pour un registre 4 bits universel.
2. Combien de multiplexeurs sont nécessaires ?
3. Simuler les opérations suivantes (état initial = 0000) :

Cycle	Mode	Entrées	État
0	-	-	0000
1	3	1101	?
2	1	0	?
3	1	0	?
4	2	1	?

4. Application : Utiliser ce registre pour effectuer des multiplications/divisions par 2.

Solution partielle : Schéma : Chaque bascule a un MUX 4 :1 en entrée sélectionnant :

- $I_0 : Q_i$ (maintien)
- $I_1 : Q_{i+1}$ (décalage droit)
- $I_2 : Q_{i-1}$ (décalage gauche)
- $I_3 : D_i$ (chargement parallèle)

Nombre de MUX : 4 MUX 4 :1 (un par bascule)

Simulation :

Cycle	Mode	Entrées	État
0	-	-	0000
1	3 (Load)	1101	1101
2	1 (Shift R)	0	0110
3	1 (Shift R)	0	0011
4	2 (Shift L)	1	0111

Application arithmétique :

- **Décalage droite** : division par 2 (logique)
- **Décalage gauche** : multiplication par 2
- Exemple : $1101_2 = 13 \rightarrow$ décalage droit $\rightarrow 0110_2 = 6$ ($\lfloor 13/2 \rfloor$)
- $0110_2 = 6 \rightarrow$ décalage gauche $\rightarrow 1100_2 = 12$ (6×2)

Exercice 8 : Compteur asynchrone 3 bits

Un compteur asynchrone (ripple counter) utilise des bascules JK en mode Toggle.

1. Dessiner le schéma d'un compteur asynchrone 3 bits ($Q_2Q_1Q_0$).
2. Établir la séquence de comptage de 000 à 111 puis retour à 000.
3. Tracer le chronogramme de l'horloge et des sorties Q_0, Q_1, Q_2 .
4. Calculer le délai de propagation total si chaque bascule a un délai de 10 ns.
5. Quelle est la fréquence maximale d'horloge si on tolère un délai de 30 ns ?
6. Expliquer pourquoi ce compteur est appelé "asynchrone".
7. Comment créer un compteur modulo-6 (comptant de 0 à 5) ?

Solution : Schéma :

$H \rightarrow$
 $JK \quad Q \rightarrow \quad JK \quad Q \rightarrow \quad JK \quad Q$

$\rightarrow \quad \rightarrow$
 $J=K=1 \quad \quad \quad J=K=1 \quad \quad \quad J=K=1$

Séquence :

Impulsion	Q_2	Q_1	Q_0	Décimal
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0

Chronogramme :

$H : \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}}$
 $Q : \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \quad (\div 2)$
 $Q : \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \quad (\div 4)$
 $Q : \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \underline{\underline{\underline{\underline{|}}}} \quad (\div 8)$

Délai total : $3 \text{ bascules} \times 10 \text{ ns} = 30 \text{ ns}$ (pire cas)

Fréquence max : $f_{max} = 1/30\text{ns} \approx 33 \text{ MHz}$

Asynchrone : Les bascules ne partagent pas la même horloge. Chaque sortie déclenche la suivante.

Modulo-6 : Détecer l'état 6 (110_2) avec une porte NAND et réinitialiser :

$$RESET = Q_2 \cdot Q_1$$

Exercice 9 : Compteur synchrone 4 bits

Dans un compteur synchrone, toutes les bascules partagent la même horloge.

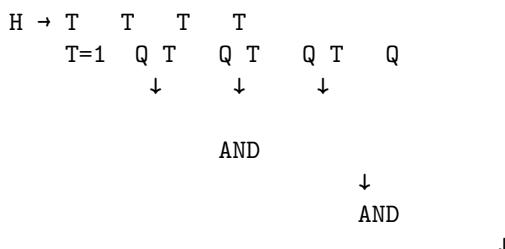
1. Pour un compteur binaire synchrone, établir la logique de contrôle :
 - $T_0 = ?$ (bascule de poids faible)
 - $T_1 = ?$
 - $T_2 = ?$
 - $T_3 = ?$
2. Dessiner le schéma logique du compteur 4 bits.
3. Vérifier la séquence pour les transitions :
 - $0111 \rightarrow 1000$ ($7 \rightarrow 8$)
 - $1111 \rightarrow 0000$ ($15 \rightarrow 0$)
4. Quel est l'avantage principal par rapport au compteur asynchrone ?
5. Calculer le délai critique (délai d'une bascule + délai logique).
6. Estimer la fréquence maximale.

Solution : Logique de contrôle (bascules T) :

$$\begin{aligned} T_0 &= 1 \text{ (toujours)} \\ T_1 &= Q_0 \\ T_2 &= Q_0 \cdot Q_1 \\ T_3 &= Q_0 \cdot Q_1 \cdot Q_2 \end{aligned}$$

Principe : Q_i bascule si tous les bits de poids inférieur sont à 1.

Schéma :



Vérifications :

- **0111 → 1000 :**
 - $Q_3 Q_2 Q_1 Q_0 = 0111$
 - $T_0 = 1, T_1 = 1, T_2 = 1, T_3 = 1$ (tous à 1)
 - Toutes les bascules basculent
 - Résultat : 1000
- **1111 → 0000 :**

- Tous les $T_i = 1$
- Toutes les bascules basculent
- Résultat : 0000

Avantage : Pas de délai cumulatif, tous changements simultanés.

Délai critique :

- Délai bascule : 10 ns
- Délai porte AND : 2 ns
- Total : 12 ns (au lieu de 40 ns pour asynchrone 4 bits)

Fréquence max : $f_{max} \approx 83$ MHz (vs 25 MHz pour asynchrone)

Exercice 10 : Compteur modulo-10 (BCD)

Un compteur décimal compte de 0 à 9 puis revient à 0.

1. Concevoir un compteur asynchrone 4 bits qui se réinitialise à 10.
2. Quelle porte logique détecter l'état 10 (1010_2) ?
3. Établir la table d'états complète (0 à 9).
4. Tracer le chronogramme des sorties $Q_3Q_2Q_1Q_0$.
5. L'état 10 apparaît-il réellement en sortie ? Pourquoi ?
6. Application : Cascader deux compteurs modulo-10 pour compter de 0 à 99.
7. Comment créer un compteur modulo-N quelconque ?

Solution : Schéma :

Compteur 4 bits
 Q Q Q Q

NAND → RESET

(détexion 1010)

Détexion de 10 : 1010_2 signifie $Q_3 = 1$ et $Q_1 = 1$

Porte : $\text{NAND}(Q_3, Q_1) = 0$ quand $Q_3 \cdot Q_1 = 1$

Table d'états :

Impulsion	Q_3	Q_2	Q_1	Q_0	Décimal
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	0	0	0	0	0 (RESET)

L'état 10 n'apparaît que brièvement (quelques nanosecondes) avant le reset.

Compteur 0-99 : Cascader deux modulo-10 :

- Premier compteur : unités (0-9)
- Sortie retenue → horloge du deuxième
- Deuxième compteur : dizaines (0-9)

Modulo-N général :

1. Utiliser compteur de $n = \lceil \log_2(N) \rceil$ bits
2. Déetecter l'état N
3. Connecter la détection au RESET

Exercice 11 : Compteur up/down

Un compteur bidirectionnel peut compter en avant ou en arrière.

1. Modifier un compteur synchrone 3 bits pour ajouter une entrée UP/DOWN :
 - UP/DOWN = 1 : comptage croissant
 - UP/DOWN = 0 : décomptage
2. Établir la logique de contrôle modifiée pour les entrées T_i .
3. Simuler la séquence (état initial = 100) :

Impulsion	UP/DOWN	État
0	-	100
1	1	?
2	1	?
3	0	?
4	0	?
5	0	?

4. Comment détecter un débordement (overflow/underflow) ?
5. Application : Utiliser ce compteur comme minuteur programmable.

Solution partielle : Logique de contrôle :

- En mode UP : T_i bascule si tous les bits inférieurs = 1
- En mode DOWN : T_i bascule si tous les bits inférieurs = 0

Équations :

$$\begin{aligned}
 T_0 &= 1 \\
 T_1 &= (UP \cdot Q_0) + (\overline{UP} \cdot \overline{Q_0}) \\
 T_2 &= (UP \cdot Q_0 \cdot Q_1) + (\overline{UP} \cdot \overline{Q_0} \cdot \overline{Q_1})
 \end{aligned}$$

Simulation :

Impulsion	UP/DOWN	État	Décimal
0	-	100	4
1	1 (UP)	101	5
2	1 (UP)	110	6
3	0 (DOWN)	101	5
4	0 (DOWN)	100	4
5	0 (DOWN)	011	3

Détection débordement :

- **Overflow** : 111 → 000 en mode UP
 - **Underflow** : 000 → 111 en mode DOWN
 - Signal = changement de Q_{MSB}
- Minuteur : Charger une valeur, compter en DOWN jusqu'à 0, générer une interruption.

Exercice 12 : Machine à états - DéTECTEUR de séQuence

Concevoir une machine à états finis qui détecte la séquence "101" dans un flux binaire.

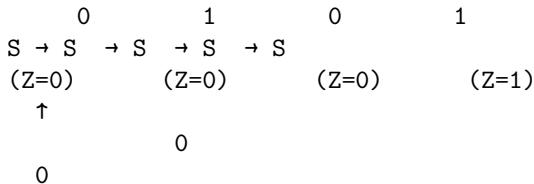
Spécifications :

- Entrée : X (bit d'entrée série)
- Sortie : Z (vaut 1 quand "101" est détecté)
- Détection avec chevauchement (ex : "10101" donne 2 détections)

1. Dessiner le diagramme d'états (Moore ou Mealy).
2. Établir la table d'états.
3. Déterminer le nombre de bascules nécessaires.
4. Choisir un encodage des états.
5. Établir les tables de vérité pour :

- Les entrées des bascules (état suivant)
 - La sortie Z
6. Simplifier les équations avec des K-maps.
 7. Dessiner le circuit complet.
 8. Tester avec la séquence d'entrée : $X = 10101100$

Solution partielle : Diagramme d'états (Moore) :



États :

- S_0 : État initial (rien détecté)
- S_1 : "1" détecté
- S_2 : "10" détecté
- S_3 : "101" détecté (sortie $Z=1$)

Table d'états :

État actuel	X	État suivant	Z
S_0	0	S_0	0
S_0	1	S_1	0
S_1	0	S_2	0
S_1	1	S_1	0
S_2	0	S_0	0
S_2	1	S_3	0
S_3	0	S_2	1
S_3	1	S_1	1

Nombre de bascules : 4 états \rightarrow 2 bascules ($2^2 = 4$)

Encodage :

- $S_0 = 00$
- $S_1 = 01$
- $S_2 = 10$
- $S_3 = 11$

Test avec $X = 10101100$:

Cycle	X	État	Z
0	-	S_0	0
1	1	S_1	0
2	0	S_2	0
3	1	S_3	0
4	0	S_2	1 \leftarrow détection
5	1	S_3	0
6	1	S_1	1 \leftarrow détection
7	0	S_2	0
8	0	S_0	0

2 détections dans "10101100"

Exercice 13 : Horloge numérique

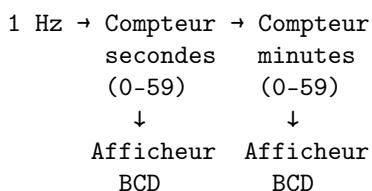
Concevoir un système d'horloge simple (secondes et minutes).

Spécifications :

- Compteur de secondes : 0-59 (modulo-60)
- Compteur de minutes : 0-59

- Horloge de base : 1 Hz
- Affichage BCD (2 chiffres par compteur)
- 1. Dessiner le schéma bloc global.
- 2. Comment créer un compteur modulo-60 ?
 - Avec deux compteurs modulo-10 ?
 - Avec un compteur modulo-6 et un modulo-10 ?
- 3. Comment générer l'impulsion pour le compteur de minutes ?
- 4. Ajouter une fonction de remise à zéro globale.
- 5. (Bonus) Ajouter des compteurs d'heures (modulo-24).
- 6. Estimer le nombre total de bascules nécessaires.

Solution partielle : Schéma bloc :



Modulo-60 : Deux approches

Approche 1 : Unités (0-9) + Dizaines (0-5)

- Compteur unités : modulo-10
- Quand unités = 9 → impulsion vers dizaines
- Compteur dizaines : modulo-6
- Quand dizaines = 5 ET unités = 9 → reset global

Approche 2 : Compteur 6 bits avec détection de 60

- Compteur binaire : 000000 à 111011 (0 – 59)
- Déetecter 111100 (60) et réinitialiser

Impulsion minutes : Signal de retenue quand secondes 59 → 0

Circuit de retenue :

$$Carry = (S_{dizaines} = 5) \cdot (S_{units} = 9)$$

Remise à zéro : Signal RESET global connecté à tous les compteurs.

Heures (modulo-24) :

- Compteur unités : modulo-10
- Compteur dizaines : modulo-3 (mais reset à 24)
- Détection : 24 → 0

Nombre de bascules :

- Secondes : 8 bascules (2×4 bits BCD)
- Minutes : 8 bascules
- Heures : 8 bascules
- Total : 24 bascules minimum

Exercice 14 : Projet - Feu de circulation

Concevoir un contrôleur de feu de circulation avec machine à états.

Spécifications :

- 2 feux : Nord-Sud (NS) et Est-Ouest (EO)
- Séquence : NS vert → NS jaune → EO vert → EO jaune → (boucle)
- Durées : Vert 30s, Jaune 5s
- Capteur voiture : peut modifier la durée

États des feux :

- État 0 : NS vert, EO rouge (30s)
- État 1 : NS jaune, EO rouge (5s)
- État 2 : NS rouge, EO vert (30s)
- État 3 : NS rouge, EO jaune (5s)

1. Dessiner le diagramme d'états.
2. Établir la table d'états complète.
3. Concevoir le système de temporisation (compteur).
4. Définir les équations pour les sorties (6 LEDs : R, J, V × 2).
5. Comment intégrer le capteur de voiture ?
6. Simuler une séquence complète.
7. (Bonus) Ajouter un mode piéton avec bouton.

Solution partielle : Diagramme d'états simplifié :



Table d'états :

État	Durée	Suivant	NS			EO		
			R	J	V	R	J	V
S_0	30s	S_1	0	0	1	1	0	0
S_1	5s	S_2	0	1	0	1	0	0
S_2	30s	S_3	1	0	0	0	0	1
S_3	5s	S_0	1	0	0	0	1	0

Temporisation :

- Compteur modulo-30 pour les états verts
- Compteur modulo-5 pour les états jaunes
- Ou compteur unique avec comparateurs

Équations des sorties (encodage S_1S_0) :

$$\begin{aligned}
 NS_V &= \overline{S_1} \cdot \overline{S_0} \\
 NS_J &= \overline{S_1} \cdot S_0 \\
 NS_R &= S_1 \\
 EO_V &= S_1 \cdot \overline{S_0} \\
 EO_J &= S_1 \cdot S_0 \\
 EO_R &= \overline{S_1}
 \end{aligned}$$

Capteur voiture :

- Si capteur EO actif en état S_0 : réduire durée à 15s
- Si pas de capteur NS en S_2 : réduire à 15s aussi

Mode piéton :

- Bouton → demande passage
- Après état jaune : insérer état "piétons" (tous rouges)
- Durée : 10s
- Puis reprendre cycle normal